PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS

MAY 1971 - NUMBER 8

BY

R:H. BUSEK

JHRP

JOINT HIGHWAY RESEARCH PROJECT

PURDUE UNIVERSITY AND
INDIANA STATE HIGHWAY COMMISSION

71-8

Final Report

PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS

TO: J. F. McLaughlin, Director
Joint Highway Research Project

May 11, 1971

File No.: 7-4-17

FROM: H. L. Michael, Associate Director
Joint Highway Research Project

Project No.: C-36-56Q

The attached Final Report titled "Practical Optimization of Steel Highway Bridge Beams" is submitted for acceptance as fulfilling the objectives of the Plan of Study "Optimum Design of Continuous Highway Bridge Girders" approved by the Advisory Board on February 17, 1970. Mr. Robert H. Busek, Graduate Assistant In Research on our staff, authored the report and conducted the research under the direction of Professors J. T. Gaunt and A. D. M. Lewis.

A program for optimization of a rolled section highway bridge girder is presented with either minimization of weight or cost. Either composite or non-composite design theory can be used. Two published design examples were solved by the computer and the results are reported and compared to the published results.

The research was financed from JHRP funds and is submitted with a request that comments on the report are encouraged. A report of application of the results by the ISHC would certainly be welcome.

Respectfully submitted,

Harold L. Michael

Harold L. Michael
Associate Director

HLM:ms

cc: F. L. Ashbaucher
W. L. Dolch
W. H. Goetz
W. L. Grecco
M. J. Gutzwiller
G. K. Hallock
M. E. Harr
R. H. Harrell
M. L. Hayes
E. M. Mikhail
R. D. Miles
J. W. Miller
C. F. Scholer
M. B. Scott
W. T. Spencer
N. W. Steinkamp
H. R. J. Walsh
K. B. Woods
E. J. Yoder

Final Report

PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS

by

Robert H. Busek
Graduate Assistant in Research

Joint Highway Research Project

File: 7-4-17
Project: C-36-56Q

Purdue University
Lafayette, Indiana
May 11, 1971

Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

http://www.archive.org/details/practicaloptimiz7108buse

ACKNOWLEDGMENTS

The author expresses his appreciation and thanks to Professor John T. Gaunt and Professor Albert D. M. Lewis, who were co-chairmen of his advisory committee, for their help in selecting this topic and for their guidance, leadership and comments throughout the course of the research project.

The financial support of the Joint Highway Research Project is gratefully acknowledged. The guidance and help of Professor Harold L. Michael, the Associate Director of the Project, is appreciated.

The confidence, expressed in him by Neil H. Bettigole, is acknowledged by the author. Many thanks go to Linda and Frances for their support, help and friendship during the past two years at Purdue.

To his family, who, although they remained many miles away, always expressed their love and concern, as well as an interest in the research, the author is eternally grateful.

TABLE OF CONTENTS

Page

LIST OF TABLES v
LIST OF FIGURES. vi
LIST OF SYMBOLS. vii
ABSTRACT x
CHAPTER I - INTRODUCTION. 1

1.1 Structural Optimization. 1
1.2 Background Review. 3
1.3 Scope of the Investigation. 5

CHAPTER II - GIRDER ANALYSIS 7

2.1 Design Loads 7
2.2 Analysis Theory 10
2.3 Analysis Program 19
2.4 Load Subroutine 27

CHAPTER III - GIRDER DESIGN. 32

3.1 Girder Design Theory. 32
3.2 Fatigue Design. 38
3.3 Design Details. 41
3.4 Design Program. 45

CHAPTER IV - PROBLEM OPTIMIZATION. 54

4.1 Optimization Theory 54
4.2 Objective Function 61
4.3 Constraints. 63

CHAPTER V - COMPUTER PROGRAM 69

5.1 Description of the MAIN Program 69
5.2 Sample Problems 78

Page

CHAPTER VI - SUMMARY AND CONCLUSIONS. 85

6.1 Summary 85
6.2 Recommendations for Further Research 86

BIBLIOGRAPHY. 88

APPENDICES

APPENDIX A - SUBROUTINE DESCRIPTIONS. 91

APPENDIX B - SELECTED PROGRAM NOMENCLATURE. 95

APPENDIX C - PROGRAM DATA 108

READIN Subroutine 108
Method of Data Input 113
Input Data Samples 123

APPENDIX D - COMPUTER PROGRAM LISTING 127

LIST OF TABLES

Table Page

2.1 AASHO Load Designations and Parameters. 8

3.1 Values of the Modular Ratio 36

3.2 Fatigue Constants. 40

5.1 Composite Section Properties 79

5.2 Unit Costs 79

5.3 Costs and Weights for Problem No. 1. 80

5.4 Costs and Weights for Problem No. 2. 82

Appendix
Table

C-1 Input Data for Sample Data Problem No. 1 124

C-2 Input Data for Sample Data Problem No. 2 126

LIST OF FIGURES

Figure Page

2.1 Determination of Compatibility Equations. . . . 15

2.2 Sample Girder for Equation Development . . . 18

2.3 Analysis Program Flowchart 20

3.1 Standard Composite Girder Cross Section . . . 35

3.2 Facia Girder Composite Cross Section 35

3.3 Design Program Flowchart 46

5.1 Flowchart for the MAIN Program 70

5.2 Optimum Design for Problem No. 1 81

5.3 Optimum Design for Problem No. 2 83

LIST OF SYMBOLS

$a(x)$ = the influence line function

a_{ij} = the influence coefficient at point i

A_{pl} = the area of a cover plate

A_r = the area of the longitudinal steel reinforcement

A_{web} = the area of the steel section web

b = the width of the steel section flange

c = the distance to the extreme steel fiber

E_c = the modulus of elasticity of concrete

E_s = the modulus of elasticity of steel

f = the longitudinal flexural stress

f_{all} = the allowable flexural stress

f'_c = the 28-day concrete strength

f_{ij} = the flexibility coefficient

F_a = the allowable column stress for the stiffeners

F_b = the allowable bending stress

F_r = the allowable stress reduced for fatigue

F_u = the minimum tensile strength of steel

F_v = the allowable weld metal stress

F_y = the yield point stress of steel

$g(x)$ = the distributed load function

I_i = the moment of inertia of element i

I_{hmc} = the moment of inertia of the high modulus concrete section

I_{lmc} = the moment of inertia of the low modulus concrete section

I_{st} = the moment of inertia of the steel section

ℓ = the unsupported length of the compression flange

ℓ_i = the length of the subelement i

$L_{cp_{min}}$ = the minimum cover plate length

L_w = the length of a weld

M_{DL} = the moment due to the dead load

M_i = the moment at section i

M_{LL} = the moment due to the live load

M_{SLL} = the moment due to the superimposed live load

M_{tot} = the total moment at a given section

n = the modular ratio for concrete

n_c = the number of shear connectors per section

P_j = the concentrated load at point j

Q = the statical moment of the area above or below a section, taken about the neutral axis

R = the algebraic ratio of the minimum stress to the maximum stress

S_r = the range of horizontal shear per inch

S_u = the ultimate strength of a shear connector

t = the required weld size

$t_{cp_{max}}$ = the maximum cover plate thickness

t_f = the flange thickness of the rolled section

t_w = the web thickness of the rolled section

v = the unit shear stress

V = the external vertical shear

V_r = the range of external vertical shear due to the live load plus impact

w = the magnitude of a uniformly distributed load

W_{pl} = the width of a cover plate

y_{lmc} = the extreme fiber distance for the low modulus composite section

y_{hmc} = the extreme fiber distance for the high modulus composite section

y_{st} = the extreme fiber distance for the steel section

Z_r = the allowable shear per connector

ABSTRACT

Busek, Robert Henry. M.S.C.E., Purdue University, June 1971. PRACTICAL OPTIMIZATION OF STEEL HIGHWAY BRIDGE BEAMS. Major Professors: J. T. Gaunt and A. D. M. Lewis.

A program for the optimization of a rolled section highway bridge girder is presented. The girder is designed by minimizing either the weight or the cost. The cost function used is a representative model of the total cost of the girder, including both material and fabrication costs. The function presented is a sample and can be altered if a different cost function is more appropriate.

The program produces the optimum design of a rolled section girder, using either composite or noncomposite design theory. The required cover plates are determined, using both static and fatigue stress conditions. The fatigue stresses are based on 500,000 cycles of stress. If a design is composite, 7/8-in. shear connector studs are provided in the positive moment region. No connectors are placed in the negative moment region. All design theory, analysis theory, and optimization constraints are based on the 1969 AASHO bridge specifications. The method of influence lines is used in the analysis.

The exhaustive search technique is used as the main method of optimization. The cover plate thicknesses are

determined by using the method of interval halving, and the locations of the cover plate splices are determined by using dynamic programming.

Computer solutions for two published design examples were obtained and compared to the published results.

CHAPTER I

INTRODUCTION

1.1 Structural Optimization

Optimization is the process of obtaining "the best" solution to a problem based on a given criterion. Structural optimization, therefore, is the process of obtaining the "optimum" solution to a structural problem. This optimum is obtained by using analysis and design theory and an optimum-seeking method suited to the problem.

Structural optimization has only recently been approached systematically. One of the reasons for this is that the computer is a relatively new tool for the engineer. When doing an optimization, there are a large number of calculations which must be completed. Performance of these calculations by hand would be very time consuming. The computer gives the engineer a quick and fairly inexpensive method of calculating quantities and making decisions based on these calculations.

Another reason for the recent prominence of structural optimization is the rising cost of construction. Since the costs of material and labor have risen steadily over the past years, and since the trend for the future seems to be in the same direction, the engineer must place more emphasis

on the economics of design. By using optimization procedures, the designer may actually reduce construction costs by minimizing the total cost of the structure.

The formulation of the optimization problem depends on the type of structure being designed. For any problem, the design variables, the design constraints and the criterion (or the objective function) of the optimization must be defined. It is the values of the design parameters which are desired so that the value of the objective function is a minimum and the constraints are not violated.

The design parameters required depend on the type of optimization problem and on the theoretical equations used. The specification used to control the design specifies the necessary constraints for the problem. In most problems, both equality and inequality constraints are used.

The objective function can be any desirable measure of the effectiveness of the design. Some of the measures which have been used are weight, cost, cost - benefit ratios and reliability. For the highway bridge girder, minimum weight has been used for many years, because of the relationship which exists between weight and cost. In this optimization problem, however, the cost function used is more realistic because it includes the fabrication costs.

The general optimization problem can be expressed in the following mathematical form:

Find the variables x_i $\quad i = 1, 2, \cdots, m$

which optimize $G(x_i)$ $\quad$ 1.1

where $g(x_p) = 0 \quad p = 1, 2, \cdots, n$ $\quad$ 1.2

$h(x_k) \geq 0 \quad k = n, \cdots\cdots, m$ $\quad$ 1.3

$x_j \geq 0 \quad j = 1, 2, \cdots, m$ $\quad$ 1.4

The variable vector, x, must be determined so that the equality constraints (Equation 1.2) and the inequality constraints (Equations 1.3 and 1.4) are satisfied and the objective function, G(x), is optimized. The G function can be either maximized or minimized depending on the type of problem.

The optimum-seeking methods which are available to the structural designer are numerous. Many of the methods are general in nature and can be used for any type of optimization. There are, however, some specific methods which are best suited for particular types or formulations of problems.

1.2 Background Review

The methods of optimization can be divided into direct and indirect methods. The indirect methods are mathematical methods which can use the gradient of the algebraic form of the objective function to solve a given problem. Direct methods are ones which make trial solutions in some organized manner. Each solution is compared to the present optimum and a decision is then made as to the next design

step. Each of the types has its own advantages and disadvantages when used in a particular structural optimization problem.

Structural optimization problems have been approached in the past as minimum weight design problems. Drucker and Shield [1 and 2] have presented some of the general minimum weight design theory. Faulkes [3] and Heyman [4] have used this criterion in optimizing frames, and Krishman and Shetty [5 and 6] and Haug [7] have used it in elastic beam design problems. Hahn [8] extended the work to include plastic design of beams and simple structures. The assumption made in these minimum weight problems is that the cost of the girder is proportional to its weight.

Computer design methods have been developed for highway bridge design. Sturman, Albertson, Cornell and Roesset [9] developed a program called BRIDGE which can be used to make comparative studies of different types of bridges. The system can also completely design a particular structure, but it makes no attempt to optimize the design. Since the complete optimization of a highway bridge is a long and complicated problem, the approach so far has been to optimize the individual parts of the bridge system.

The optimization of girders has also been approached by using the minimum weight method. Edward [10] and Holt and Heithecker [11] have attempted to optimize the girder by minimizing the cross-sectional area. This in turn minimizes the weight of the structure.

The total cost optimization has been developed for both building and highway girders. Annamalai [12] uses the backtrack programming method for welded plate girders, developed by Lewis [13], to optimize the design of building plate girders based on the AISC 1969 specifications. Okuba [14] solves the girder problem by using a linear programming method.

The work by Razani, Goble and DeSantis [15, 16, and 17] on the optimum design of plate girders has led to the development of a computer-aided design system. Goble and DeSantis [18 and 19], with the aid of the Ohio Department of Transportation, developed a program referred to as Girder Automated Design - I or GAD-I. The program completely designs the constant depth plate girder and specifies all of the details of the design, including the flange thicknesses, the flange splice locations and the locations of the transverse and longitudinal stiffeners if they are required.

1.3 Scope of the Investigation

Since the GAD-I program has been developed, it has been possible to design the optimum welded plate girder for a highway bridge. This girder, however, is not necessarily the actual optimum design for the problem. There is a possibility that either a composite or noncomposite rolled-section girder may be more economical than the plate girder. This is particularly true if the spans of the girder are relatively short.

In order to determine the true optimum design, a computer-aided design system for the rolled-section bridge girder has been developed. With this program, a given problem can be designed with a plate girder and a rolled-section girder and the true optimum can be chosen directly.

In the present study, the American Association of State Highway Officials bridge code [20] is used to control the design of a highway bridge girder. The cross section of the girder is limited to the 36-in. and 33-in. wide-flange rolled sections. The design can be either composite or noncomposite and can have cover plates if they are required. The problem is limited to a maximum of four continuous spans designed for all types of static and fatigue highway loads. The fatigue conditions are based on 500,000 cycles of stress. If the design is composite, then 7/8-in. diameter shear connector studs are provided at the proper spacing. The method of exhaustive search is used in the program because of the limited number of rolled sections of proper size. Each of the eighteen possibilities is designed, using the optimum-seeking methods of interval halving and dynamic programming to determine the required cover plates.

CHAPTER II

GIRDER ANALYSIS

2.1 Design Loads

The first problem that confronts the designer of a particular structure is to determine the required design-load conditions for that structure. These conditions usually vary with the size, purpose and location of the structure. The actual conditions are often designated by design specifications and can be found in design codes.

The design of a highway bridge is controlled by the American Association of State Highway Officials code [20]. This code, known as the AASHO code, states that all of the girders in a bridge structure must be designed for the largest truck load and the corresponding equivalent lane load which the bridge must sustain. The code designates the loads by the size of the truck being considered, i.e., HS20. Once the required designation is chosen, the actual loading parameters can be determined. These parameters include the axle loads, the axle spacings, the equivalent lane loads and the corresponding concentrated loads. Table 2.1 contains a summary of the load parameters for all of the AASHO truck designations.

Table 2.1: AASHO Load Designations and Parameters

	Truck Loadings						Equivalent Lane Loading		
	Axle Loads			Axle Spacings			Uniform	Concentrated	
Des.	1	2	3	1 - 2	Min 2 - 3	Max 2 - 3		Moment	Shear
H10	4^k	16^k		14'			$.32^k/'$	9.0^k	13.0^k
H15	6^k	24^k		14'			$.43^k/'$	13.5^k	19.5^k
H20	8^k	32^k		14'			$.64^k/'$	18.0^k	26.0^k
HS15	6^k	24^k	24^k	14'	14'	30'	$.48^k/'$	13.5^k	19.5^k
HS20	8^k	32^k	32^k	14'	14'	30'	$.64^k/'$	18.0^k	26.0^k

If the bridge is being designed as part of the Interstate Highway System, the girders must also be designed for a specified interstate loading condition. This loading condition is based on the dimensions and weight of the largest military vehicle which would use the structure during a national emergency. The loading consists of two, 24-kip concentrated loads with an axle spacing of 4 ft. The effect of this loading must be calculated in order to determine if it governs the girder design.

Another possible girder loading condition is produced by pedestrian pathways. If the bridge is supplied with sidewalks, then the sidewalk loads must be calculated. According to the AASHO code, the following set of formulas is used in calculating the sidewalk load:

$$P = 85 \qquad \text{for } L_s \leq 25 \text{ ft}$$

$$P = 60 \qquad \text{for } 25 \text{ ft} < L_s \leq 100 \text{ ft}$$

$$P = \left(30 - \frac{3000}{L_s}\right) \times \left(\frac{55-W}{50}\right) \qquad \text{for } L_s > 100 \text{ ft} \qquad 2.1$$

where P = the load in pounds per square foot.

L_s = the loaded length of sidewalk in feet.

W = the width of sidewalk in feet.

The total sidewalk load is distributed over all of the girders in the structure. Although the moments due to this loading condition are usually small, they are included to provide a complete analysis.

Besides the above loads, which are considered to be the live loads for the structure, there are also dead loads which must be considered. Dead loads include the weight of the steel girder, the weight of the deck slab and the weight of the bridge railing, light fixtures, and similar loads placed on the constructed bridge. If the design of the girder is noncomposite, then all of these loads are combined in one dead load condition. If the design is composite, however, the loads are placed into two groups because they act on different composite sections. The first group is considered the dead load and includes the weight of the slab and the steel girder. The second group is the superimposed dead load (sometimes referred to as the long-term live load). This group contains dead loads applied to the composite girder after the concrete has reached its maximum strength

capacity. An example of a superimposed dead load is the weight of the bridge railings and the light fixtures.

2.2 Analysis Theory

Once the loads which act on the structure have been defined, the stresses caused by these loads must be determined. The design loads must be placed on the structure so that maximum design conditions are developed at the critical points in the structure. The determination of these design values constitutes a complete analysis of the structure being designed.

Since the loading conditions, as shown above, for the highway bridge girder are complex in nature, the method of analysis which is most efficient is the method of influence lines. This method enables the structure to be loaded with a large number of loading conditions and the controlling maximum to be chosen from these conditions.

An influence line is a graph showing the variation in a particular function at a point with respect to the position of a unit load. The ordinates of the influence line are referred to as the influence coefficients. The ordinate at point j for the function F at point i, usually called the influence coefficient a_{ij}, is defined as the value F_i due to a unit load at point j [21]. With this definition various theorems can be developed pertaining to the use of influence lines. Norris and Wilbur [22] describe the following four basic influence line theorems:

Theorem 1 - The maximum value of a function due to a single concentrated load is found by placing the load over the maximum influence coordinate.

Theorem 2 - The value of a function due to a single concentrated load equals the product of the load and the influence coefficient at the load location.

Theorem 2 is stated mathematically as

$$F_i = P_j \times a_{ij} \tag{2.2}$$

From Theorem 1, the maximum value of F_i, $F_{i_{max}}$, is found when the single concentrated load is placed at the point where a_{ij} is the maximum. By using superposition, the value of the function F due to a series of concentrated loads can be found from Equation 2.3.

$$F_i = \sum_{j=1}^{m} P_j \times a_{ij} \tag{2.3}$$

where m is the number of loads.

Theorem 3 - The maximum value due to a distributed load occurs when the structure is loaded over those portions which have influence coefficients with the same sign as the character of the function desired.

Theorem 4 - The value of the function due to a distributed load is the product of the load

magnitude and the net area under the portion of the influence line loaded.

Theorem 4 is stated mathematically as

$$F = \int_{r}^{s} g(x) \cdot a(x)dx \qquad 2.4$$

where r and s are the load limits.

The derivation of Equation 2.4 can be found in the structural analysis texts by Norris and Wilbur [22] and by Shedd and Vawter [23].

There are two simplifications that can be made to Equation 2.4 when dealing with highway structures. In all cases, the distributed load, g(x), can be replaced by a constant representing the uniformly distributed loading condition. The uniform load is usually represented by w. The second simplification deals with the influence line function, a(x). Since this function is usually fairly complex in nature and too difficult to obtain, the coefficients are determined at a number of discrete points and the values in between are found using linear interpolation. The value of the function F, due to a uniform loading condition, over a piecewise continuous influence line, can be determined by using Equation 2.5.

$$F_i = \sum_{k=1}^{n} \frac{w}{2}\left(a_{i,k} + a_{i,k+1}\right)\ell_k \qquad 2.5$$

where n is the number of elements.

With the theorems and the equations expressed above, any effect of a loading condition can be obtained, provided the influence coefficients can be calculated. The influence line is developed by computing the required influence coefficients for each position of a unit concentrated load. The process is quite simple for statically determinate girders, but becomes much more complicated for indeterminate or continuous girders.

For the statically determinate problem, all that is required to solve for any influence coefficient are the equilibrium equations. The unit load is placed on the structure and the reactions are determined using the equations shown as Equation 2.6:

$$\sum M_{\text{left end}} = 0$$

$$\sum F_{\text{vertical}} = 0 \qquad 2.6$$

Once the reactions are found, the influence coefficient for any function can be determined.

If the girder being designed is statically indeterminate, then equations of compatibility are required. One compatibility equation is needed for each redundant reaction. These equations refer to the requirement that all the displacements throughout the structure are consistent. Willems and Lucas [24] note that the compatibility conditions require that the displacements at supports must be externally

satisfied. In the particular case of unyielding supports, the total displacement must be equal to zero.

The compatibility equations can be developed with reference to Figure 2.1. The structure is made statically determinate by removing the interior supports (Figure 2.1b). With the determinate structure defined, the deflections at the interior support points, due to the external loads on the simple girder, can be determined. Next the flexibility coefficients are determined (Figure 2.1c and d) and the equations can be formed:

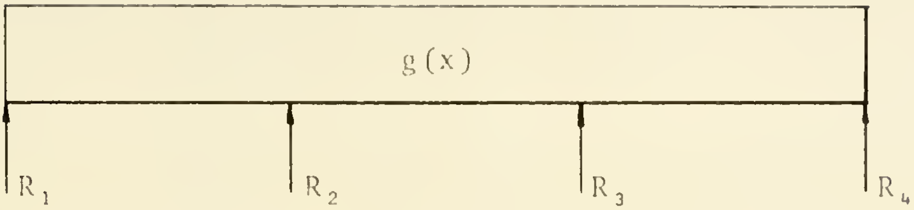
$$R_2 f_{22} + R_3 f_{32} - D_2 = 0$$
$$R_2 f_{23} + R_3 f_{33} - D_3 = 0 \qquad 2.7a$$

In matrix form, these equations may be written

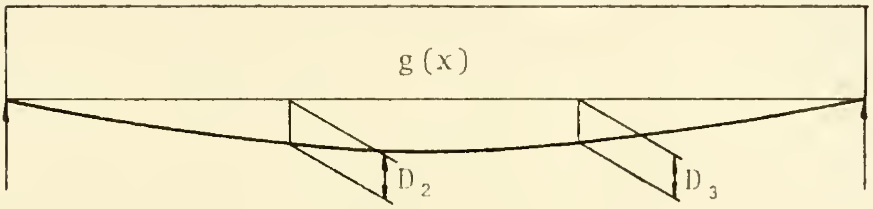
$$\begin{bmatrix} f_{22} & f_{32} \\ f_{23} & f_{33} \end{bmatrix} \begin{bmatrix} R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} D_2 \\ D_3. \end{bmatrix} \qquad 2.7b$$

or $$[\,F\,] \quad [\,R\,] = [\,D\,] \qquad 2.8$$

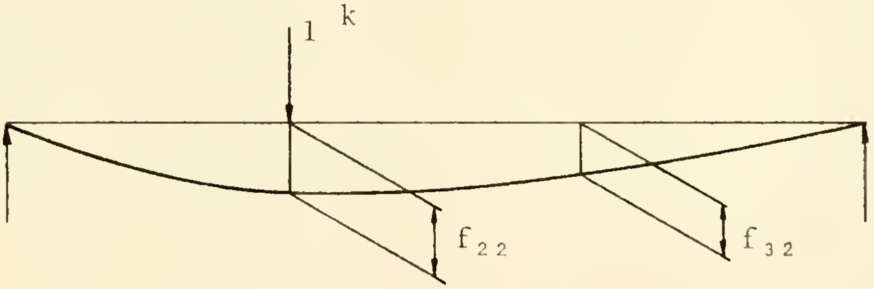
The F matrix is called the flexibility matrix and is independent of the external loads on the structure. The coefficients of this matrix represent the deflections at the interior support locations due to unit loads at the interior support locations. These deflections can be found by using the moment-area theorems.



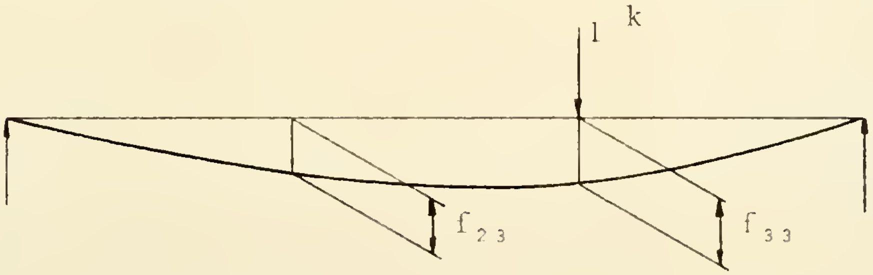
2.1 a: Indeterminate Structure



2.1 b: Determinate Structure



2.1 c: Unit Load at Point 2



2.1 d: Unit Load at Point 3

Figure 2.1: Determination of Compatibility Equations

The moment-area theorem used to determine the deflection is called the second moment-area theorem. The theorem states that the deflection at point 1, relative to a tangent at point 2, is equal to the moment of the area under the bending moment diagram, between points 1 and 2, about point 1 divided by the flexural rigidity of the girder. Pippard and Baker [25] give the following mathematical form of the theorem:

$$D = \frac{1}{EI} \int_{1}^{2} Mxdx \qquad 2.9$$

If the moment of inertia varies along the girder and the moment diagram is given as a piecewise continuous curve, varying linearly between the analysis points, then Equation 2.9 takes the following form:

$$D = \frac{1}{E} \sum_{i=1}^{n} \frac{1}{2}\left(M_i + M_{i+1}\right) \frac{\ell_i}{I_i} x_i \qquad 2.10$$

where x_i is the distance from the center of the element to the center of moments.

Once the required deflections are determined and the compatibility equations, 2.7a, are defined, the values of the redundant reactions can be determined. With these reactions and the equilibrium equations, 2.6, the remaining reactions can be calculated.

In order to develop the influence line, the unit load must be located at each point along the structure. This means that there would be NA different matrix equations of

the type shown in Equation 2.8, where NA is the number of analysis points. Since the flexibility matrix, however, does not change with the location of the external load, the only part of the equation that will vary is the D matrix. For this reason it is possible to set up one matrix of NA columns for all the positions of the unit load. Also by adding two more rows, which represent the equilibrium equations, all of the reaction influence coefficients can be determined at the same time. Shown in Figure 2.2 is a three-span girder with seven analysis points. The matrix equation required to solve for the reaction influence coefficients is

$$[\ CC\ ]\ [\ R\ ] = [\ RIL\ ] \qquad 2.11$$

where

$$[\ CC\ ] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & L_1 & L_2 & L_3 \\ 0 & d_{33} & d_{53} & 0 \\ 0 & d_{35} & d_{55} & 0 \end{bmatrix}$$

and

$$[\ RIL\ ] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \end{bmatrix}$$

This equation is the same as the one which appears in the analysis development by Goble and DeSantis [18].

The solution to this matrix equation can be approached in a number of ways. The most efficient method is one in which the final solution for the reaction influence lines

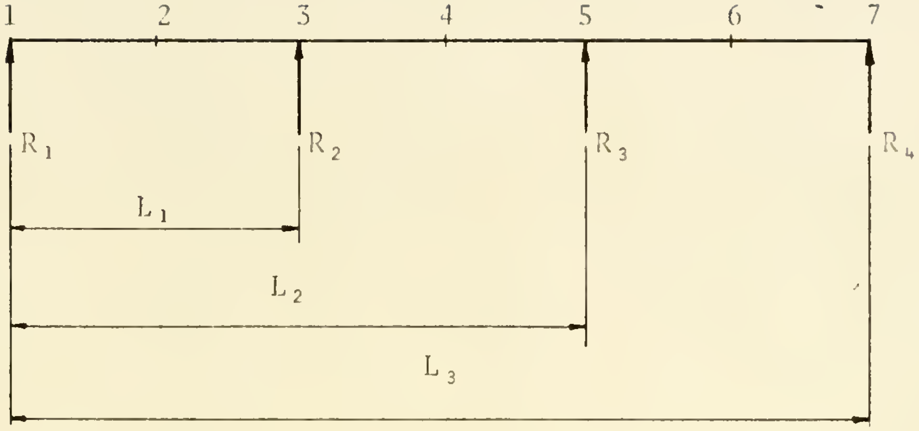


Figure 2.2: Sample Girder for Equation Development

is found in the RIL matrix. By using a Gaussian search technique, the CC matrix is reduced to an identity matrix by simple row and column operations common to matrix algebra. By repeating the same operations on the RIL matrix, the reaction influence coefficients are determined and stored in the RIL matrix. A complete description of the Gaussian search technique employed in the solution is given by Goble and DeSantis [19].

With the solution for the reaction influence coefficient complete, all of the remaining influence lines for the complete analysis can be calculated by simple statics. The influence lines can then be loaded and all the design conditions for the problem can be determined. With these conditions, the girder can be designed and optimized.

All of the theorems and equations discussed above are used to determine the design reactions, moments and shears.

After a final design has been chosen, the AASHO code requires that the deflections be calculated and compared to certain allowable live-load deflections. In order to calculate the maximum deflections, the deflection influence coefficients must be determined. The amount of calculations, however, can be reduced by employing Maxwell's law of reciprocal deflections. The law says that, on a girder, the deflection at point 1 due to a unit load at point 2 is equal to the deflection at point 2 due to a unit load at point 1. This enables the deflection influence coefficients for a point j to be determined by calculating all of the deflections along the girder due to a unit load at point j. The maximum deflections can then be calculated by using the influence coefficients and the design loads. These maximums can then be compared to the allowable deflections specified by the AASHO code.

2.3 Analysis Program

The analysis portion of the computer program uses the theory described in Section 2.2 to develop the values required by the design portion of the program. A basic flowchart for the analysis program is shown in Figure 2.3. The step numbers, used in the following program description, refer to the numbers in parentheses on the flowchart.

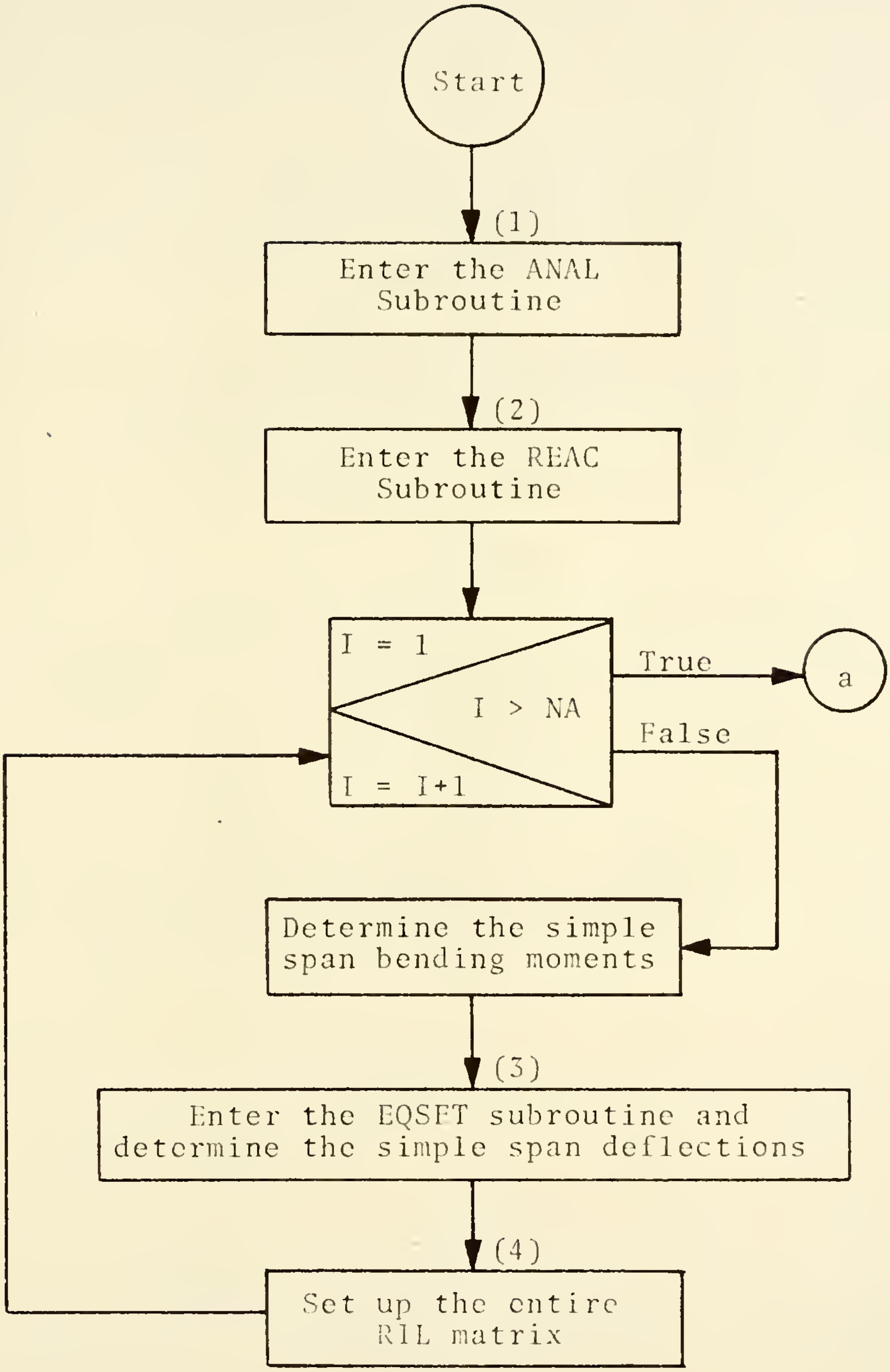

Start
(1)
Enter the ANAL Subroutine
(2)
Enter the REAC Subroutine
I = 1
I > NA
I = I+1
True
a
False
Determine the simple span bending moments
(3)
Enter the EQSET subroutine and determine the simple span deflections
(4)
Set up the entire RIL matrix

Figure 2.3: Analysis Program Flowchart

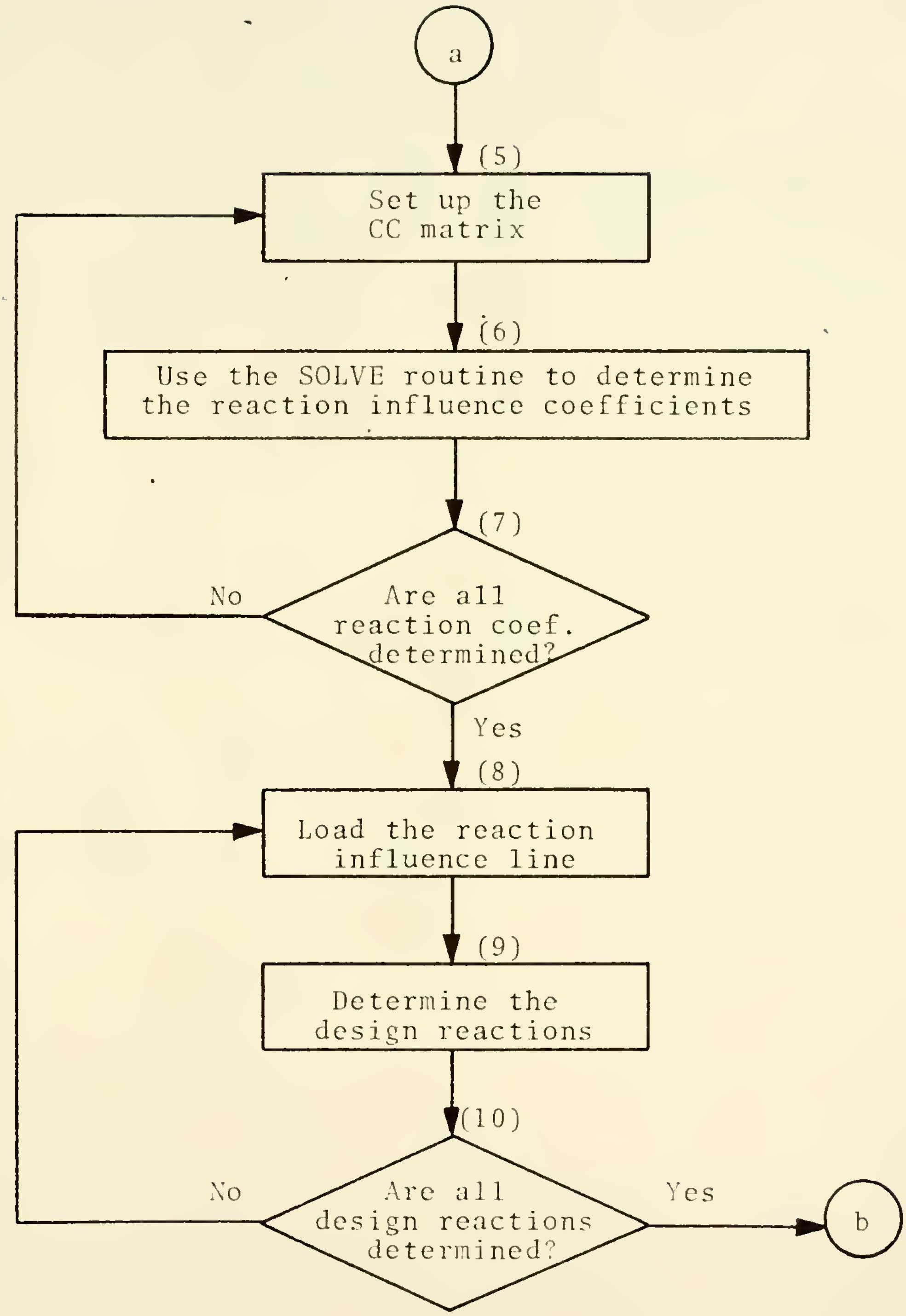

a
(5)
Set up the CC matrix
(6)
Use the SOLVE routine to determine the reaction influence coefficients
(7)
Are all reaction coef. determined?
No
Yes
(8)
Load the reaction influence line
(9)
Determine the design reactions
(10)
Are all design reactions determined?
No
Yes
b

Figure 2.3: (Cont'd.)

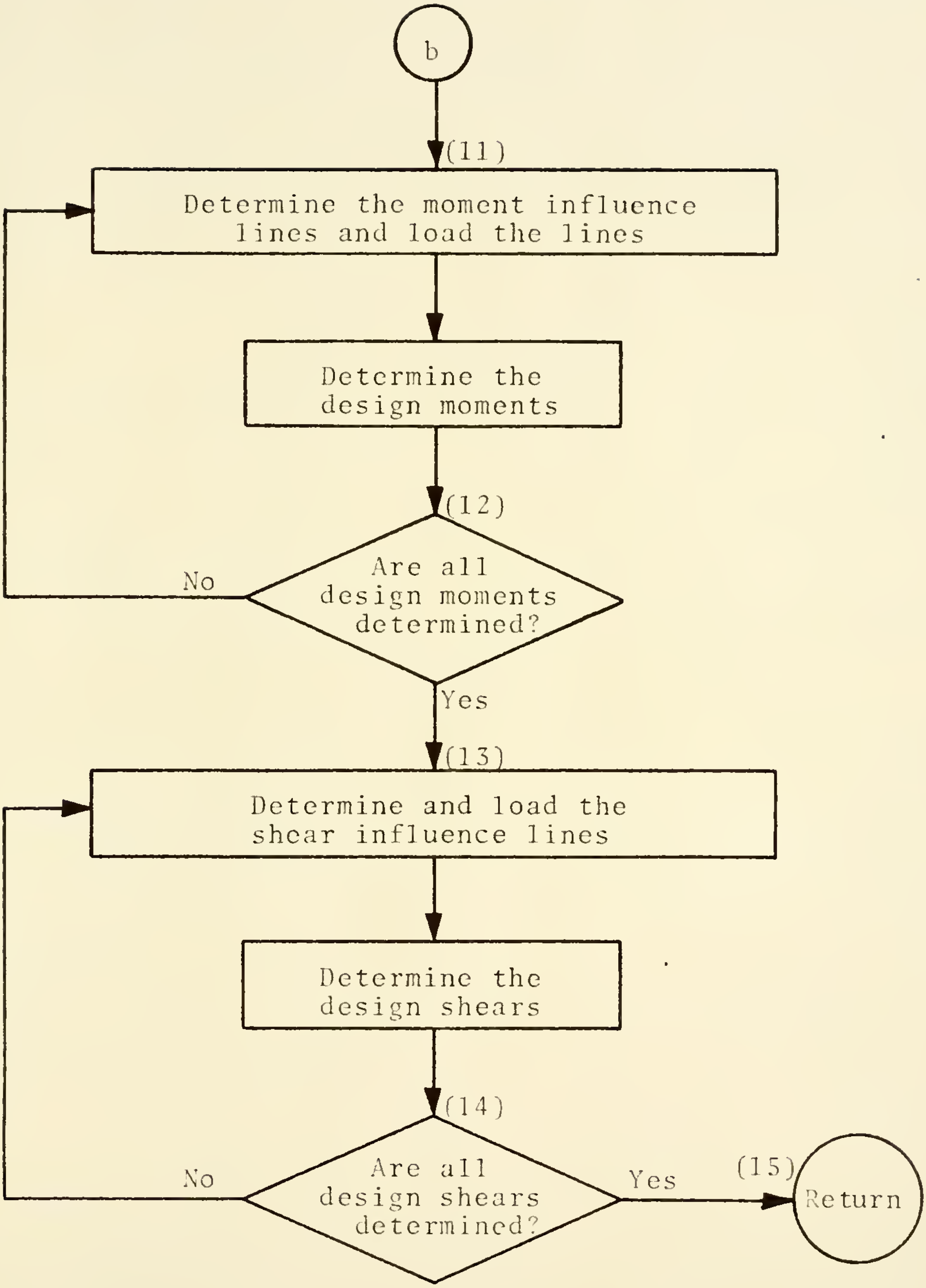

b
(11)
Determine the moment influence lines and load the lines
Determine the design moments
(12)
Are all design moments determined?
No
Yes
(13)
Determine and load the shear influence lines
Determine the design shears
(14)
Are all design shears determined?
No
Yes
(15)
Return

Figure 2.3: (Cont'd.)

Step 1

The analysis program is started by calling the ANAL subroutine. This subroutine calls on the other subroutines in the proper order required to produce a set of usable design conditions.

Step 2

The REAC subroutine is called on to solve for the reaction influence lines. The simple-span girder, defined as the problem girder with the interior supports removed, is loaded with a unit load at each analysis point, and the bending moments are determined. These simple-span bending moments are transferred to the EQSET subroutine.

Step 3

The EQSET subroutine is used to calculate the simple-span deflections at the interior support points. The value of the tangent at the left end (TALE) is determined by using Equation 2.10 and is then used in the following equation to determine the required deflections:

$$d_i = \mathrm{TALE} \times \mathrm{COOR(I)} - \mathrm{WA(I)} \qquad 2.12$$

Step 4

The RIL matrix is prepared by using the deflections obtained in step 3 and the equilibrium equations. The RIL matrix is the right-hand-side matrix of Equation 2.11.

Step 5

The coefficient matrix (CC) is prepared by using the appropriate values of the deflections found in step 3 and the coefficients of the equilibrium equations. This CC matrix corresponds to the matrix CC in Equation 2.11.

Step 6

The SOLVE subroutine is called to determine the solution to Equation 2.11. By using a Gaussian search technique, the CC matrix is reduced to an identity matrix and the reaction influence coefficients are determined and stored in the RIL matrix.

Step 7

Steps 2 thru 6 are repeated using a different moment of inertia for each subelement. By doing this, the different influence coefficients for the steel section, the low modulus concrete section, and the high modulus concrete section are determined. Each set of coefficients is stored in its proper location in the RIL matrix. Control is now passed back to the ANAL subroutine where the reaction influence coefficients are printed if IPT10 is greater than 2.

Step 8

The reaction influence line is loaded by calling on the LOAD subroutine and the values of the ten (10) loading

conditions are returned from this subroutine. Since the LOAD subroutine is used extensively in the analysis, it is described in Section 2.4.

Step 9

The design reactions (DESREA) are now determined. The dead-load reaction is set equal to STORE(1) and the superimposed dead-load reaction is set equal to STORE(10). The positive and negative live-load design reactions are equal to the maximum live-load reactions possible. These maximums are obtained by combining the sidewalk live load with the other individual live loads of the same sign until the largest value of the sums is found. If the sidewalk live load is zero, then the positive and negative design reactions are respectively equal to the largest positive and negative truck load, lane load or interstate load.

Step 10

Steps 8 and 9 are repeated for each support along the girder. The maximum number of repetitions is NS. The DESREA array is now printed out if the IPT10 parameter is greater than zero.

Step 11

The reaction influence coefficients are used to determine the moment influence line at a particular point. This influence line is then loaded by the LOAD subroutine and the ten loading conditions are determined. The design moments

moments (DESMOM) are determined in the same manner as the design reactions in step 9.

Step 12

Step 11 is repeated for each analysis point along the structure and the DESMOM array is printed if IPT10 is greater than 0. The maximum number of repetitions is NA.

Step 13

The reaction influence coefficients are used to determine the shear influence line and this line is loaded by the LOAD subroutine to calculate the values of the ten loading conditions. These conditions are used to calculate the design shears (DESSH) in the same manner as described in step 9.

Step 14

Step 13 is repeated for the locations immediately to the left and to the right of each analysis point. The maximum number of repetitions is 2NA-2. (The minus 2 is necessary because there is only one repetition made for the first and the last analysis points.) The DESSH array is printed if IPT10 is greater than 0.

Step 15

The analysis of the girder is complete and the control is returned to the main program for design.

2.4 Load Subroutine

The load subroutine is used to load any given influence line with the complex loadings required in a highway girder analysis. It should be noted that a given influence line actually consists of three different influence lines, if the design is composite. There is one line for the steel section, one for the low modulus concrete section, and one for the high modulus concrete section. All three lines are calculated in the ANAL subroutine and are transferred to the LOAD subroutine at the same time.

Step 1

The steel section influence line is set into the RO array and the properties of this line are determined by the ILPROP subroutine. The properties which are required by the LOAD routine are the positive and negative areas and the locations of the maximum and minimum ordinates.

Step 2

The unit dead load is calculated. Using the area properties of the influence line, the dead-load loading is found by using Equation 2.5 with the load extending over the entire girder. The dead-load loading is then stored in STORE(1).

Step 3

If the girder is noncomposite, then the superimposed dead-load loading (STORE(10)) is set equal to zero. If the design is composite, then the low modulus concrete section influence line is set into the RO array and the properties are found. The superimposed dead load is determined and the values of STORE(10) is calculated using Equation 2.5 with the load extending over the entire girder.

Step 4

If the design is noncomposite, then the live-load determination can be started immediately. If the design is composite, however, the high modulus concrete section influence line is set into the RO matrix and the properties are determined by the ILPROP subroutine.

Step 5

The IMPACT subroutine is called on to determine the impact factor required by the AASHO code. This factor (HT) is determined from the following formula:

$$HT = 1.0 + \frac{50}{DING-125} \qquad 2.13$$

The code places an upper limit of 1.3 on the impact factor.

Step 6

The positive lane load is found by loading the positive area with the equivalent lane load and by placing the required concentrated load over the maximum ordinate. The

load is increased by the impact factor and stored in STORE(2). The negative lane load is obtained by loading the negative area with the equivalent lane load and by placing the required concentrated load over the minimum ordinate. The AASHO code also requires a second concentrated load placed in one other span as to produce the maximum negative moment. The negative lane load is then increased by the impact factor and stored in STORE(3).

Step 7

The sidewalk live load is found by multiplying the uniform sidewalk load by the positive and negative areas respectively. The positive sidewalk load is placed in STORE(8) and the negative sidewalk load is placed in STORE(9).

Step 8

If the interstate loading is desired, the influence line is loaded with the two interstate concentrated loads which are spaced 4 ft apart. There are three loading conditions considered in the interstate loading. The first two have one load on the maximum ordinate and the second 4 ft to the right or 4 ft to the left. The third condition has the two loads centered about the maximum ordinate. The total maximum condition is increased by the impact factor and stored in STORE(4). To find the negative interstate load, the same three conditions are repeated using the

minimum ordinate as the reference. The negative interstate load is stored in STORE(5).

Step 9

The truck load analysis begins by moving the truck from left to right. A complete analysis consists of placing each of the truck's axles over the point in question and calculating the loads due to the combinations of the remaining axles.

Step 10

The first step is to place the rear axle on the analysis point and to use the maximum spacing for the front axles. The values of the influence line are determined by using the interpolation subroutine (ILINT). The resultant load is found by using Equation 2.3. This load is compared to the current maximums. (The positive maximum is found in POS and the negative maximum is found in SEG.) The calculation is repeated using the minimum spacings for the front axles.

Step 11

Next the front axle is placed on the point in question and the second axle is located the specified fixed distance away. The effect of these two axles is calculated and compared with 40 percent of the current maximums. It is assumed that if the effect of the front axles is less than 40 percent of the maximums, then the effect of the rear

axle can be neglected. If the effect of the rear axle must be calculated, the rear axle is placed at the minimum distance, at the maximum distance, and at each analysis point in between. Goble and DeSantis [18 and 19] refer to this as "wiggling the rear axles." Each total effect is compared to the current maximums and stored in POS and SEG if required.

Step 12

The final loading for a particular point is to place the second axle on the point and the first axle the specified fixed distance ahead. The effect of these two loads is calculated and compared in the same manner as in step 11. If necessary, the rear axle is wiggled and the total effects are compared to the current maximums.

Step 13

The truck load is repeated for the truck moving from right to left. The final maximums are increased by the impact factor and the positive load is stored in STORE(6) and the negative load is stored in STORE(7).

Step 14

The STORE array is now returned to the ANAL subroutine where it is used to determine the design loading conditions.

CHAPTER III

GIRDER DESIGN

3.1 Girder Design Theory

With the design conditions defined at each analysis point, the cross section can now be proportioned to carry the necessary design stresses. If the design is noncomposite, then the simple, pure bending theory can be used to determine the stresses. The equation which expresses this pure bending theory is known as the flexure formula and is shown as Equation 3.1:

$$f = M_{tot} \times \frac{c}{I} \tag{3.1}$$

or

$$f = M_{tot}/S$$

where S = the section modulus of the section and is equal to I/c.

A derivation of this formula is presented by Byars and Snyder [26] and many other mechanics textbooks.

The section used in a noncomposite design is symmetrical about the horizontal axis. This allows the top and bottom fiber stresses of the steel section to be equal. The concrete in a noncomposite design is assumed to carry no longitudinal stresses because there is no provision for the

transfer of shear between the slab and the beam. If provision is made for this transfer, then the design becomes composite.

The design of a composite bridge girder takes advantage of the high compressive strength available in the concrete slab. Some of the advantages of composite action expressed by Viest, Fountain, and Singleton [27] and by McCormac [28] are:

1. A savings in steel costs is realized,
2. The beam depth can be reduced,
3. A larger percentage of steel is in tension,
4. The deflections are reduced due to the increase in the beam stiffness,
5. A composite section can withstand a greater overload than a noncomposite design.

The one disadvantage of the composite action is the cost of providing the shear connectors. This disadvantage, however, only takes precedence over the advantages in short, lightly loaded structures.

In designing a composite girder by the AASHO code, the properties of the steel beam and slab are established on the basis of the moment of inertia of the composite section. Bresler, Lin and Scalzi [29] give the following two basic assumptions used in the analysis and design of composite beams for highway girders:

1. The slab is connected throughout the length of the girder,

2. The stress and the strain are linear across the depth of the member.

The AASHO code, unlike the American Institute of Steel Construction specification [30], does not allow any slippage for the shear connectors.

The composite design section shown in Figure 3.1 is composed of a rolled section, an effective slab section, a bottom cover plate, and a haunch. The section could also be designed without the bottom plate. It is also possible for the slab to be on one side of the girder. This type of section would be used for a facia girder and is shown in Figure 3.2.

The effective slab section is defined as a section of the bridge deck having an effective slab width and a thickness equal to the structural depth of the slab. The AASHO code defines the effective slab width of the section in Figure 3.1 as not exceeding any of the following:

1. One-fourth of the girder span,
2. The distance center-to-center of girders,
3. Twelve times the slab thickness.

If the composite section is similar to the facia girder shown in Figure 3.2, then the effective slab width should not exceed any of the following:

1. One-twelfth of the span length,
2. One-half the center-to-center distance to the adjacent girder,
3. Six times the slab thickness.

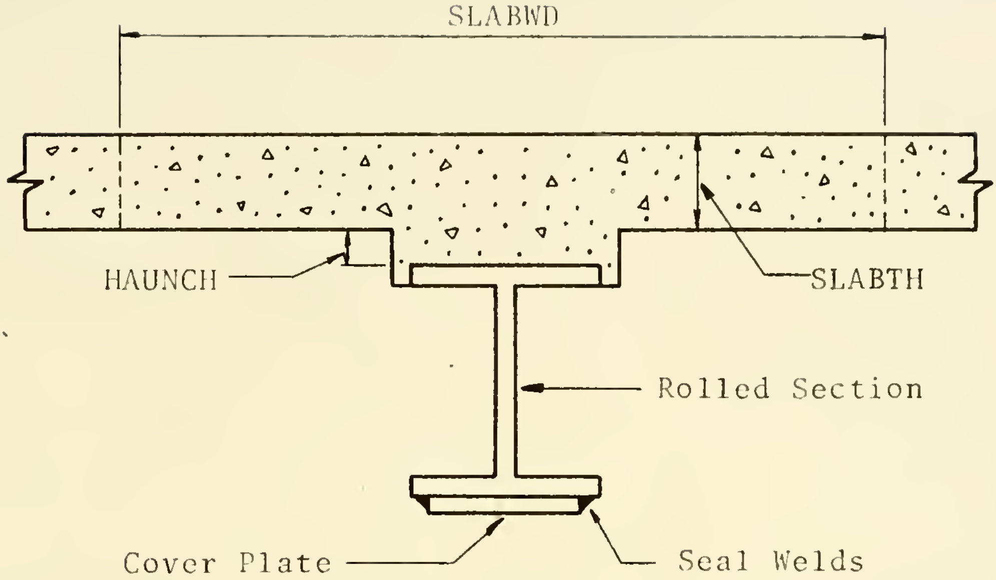


Figure 3.1: Standard Composite Girder Cross Section

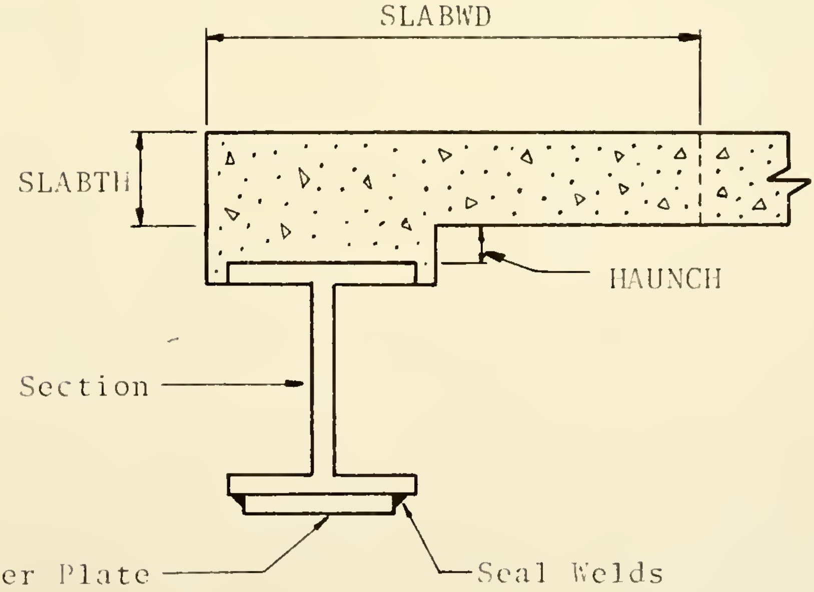


Figure 3.2: Facia Girder Composite Cross Section

The section properties of the composite section are determined by the transformed area method. This method assumes that the bond between the concrete and steel is strong enough so that the strains in the concrete and steel at the junction are equal. Due to this assumption, the relationship between the stresses at a given distance from the neutral axis is

$$f_c = \frac{1}{n} f_s \qquad 3.2$$

where

$$n = E_s/E_c$$

The factor n is called the modular ratio. The area of the concrete slab is transformed into an equivalent area of steel by dividing the effective slab width by the modular ratio. The values of the modular ratio specified by the AASHO code are shown in Table 3.1.

Table 3.1: Values of the Modular Ratio

Concrete Strength	n
2000 - 2400	15
2500 - 2900	12
3000 - 3900	10
4000 - 4900	8
5000 or more	6

The actual form of the flexure formula used in a composite design contains three terms instead of just one as in Equation 3.1:

$$f = \frac{M_{DL}\, y_s}{I_{st}} + \frac{M_{SLL}\, y_{lmc}}{I_{lmc}} + \frac{M_{LL}\, y_{hmc}}{I_{hmc}} \qquad 3.3$$

The first term in Equation 3.3 represents the stress at a point due to the dead load of the structure. This dead load is carried by the steel section alone. The properties of the steel section are independent of the slab dimensions.

The second term in the equation represents the stress due to the superimposed dead load. This load must, according to the AASHO code, be carried by the low modulus composite section. The properties of this section are calculated by using a concrete area transformed by a modular ratio equal to 3n. The increased ratio is used to account for creep effects.

The third term in the equation represents the stress due to the live load on the structure. The live load, since it is placed on the structure after the concrete has reached its maximum strength, acts on the high modulus composite section. The properties of this section are based on a concrete area transformed with a modular ratio of n.

By using Equation 3.1 for a noncomposite design and Equation 3.3 for a composite design, the steel girder cross section can be proportioned so that the resulting flexural stresses are within the allowables required by the design

code. If the design is composite, then the compressive stress in the extreme concrete fiber must also be within the accepted code allowable.

With the girder designed for flexural stresses, the shear stresses must be determined. The horizontal and vertical shear stresses at any point in the cross section can be found from Equation 3.4:

$$v = \frac{VQ}{bI} \tag{3.4}$$

where b = the width of the section where the shear stress is desired.

Byars and Snyder [26] derive this formula and also demonstrate that, for an I-shaped section, the shear stress across the web can be assumed to be uniform. Since the stress is uniform, the AASHO code states that all of the external vertical shear is carried by the steel girder web and can be calculated using Equation 3.5:

$$v = V_{tot}/A_{web} \tag{3.5}$$

For this calculation, the area of the web is equal to the product of the overall depth of the rolled section and the web thickness.

3.2 Fatigue Design

In a highway bridge girder there is a fluctuating stress level at every point because of the alternating loading and unloading of the structure due to the live

loads. This continuous fluctuation causes dislocations in the crystalline structure of the steel, and eventually causes cracks and other localized failures. This type of failure is referred to as a fatigue failure.

The fatigue failures in a structure may occur at loads which are well below the allowable static loading conditions, because the stress is continually fluctuating over a particular range. The larger the fluctuation, the worse the condition. The fatigue condition becomes extremely critical when the stress alternates between tension and compression.

The number of fluctuations, or cycles, for which a girder can be designed varies from 100,000 cycles to 2,000,000 cycles. The exact number of cycles used depends on the use and location of the bridge.

The method used in fatigue design is to reduce the allowable stresses at critical points. The AASHO code provides the following formulas for this fatigue reduction:

$$F_r = \frac{k_1 f_{ro}}{1 - k_2 R} \qquad 3.6$$

and

$$F_r = \frac{.55 F_y}{1 - \left(\frac{.55 F_y}{k_1 f_{ro}} - 1\right) R} \qquad 3.7$$

where

$$k_1 = 1.0 + \alpha\left(\frac{F_u}{58000} - 1.0\right)$$

and R = the algebraic ratio of the minimum stress to the maximum stress.

The k_2, f_{ro}, and α are the fatigue constants. These fatigue constants have been determined primarily from fatigue tests. A description of some of the tests and the results are given in the U. S. Steel bridge manual [31].

The values of the fatigue constants depend on the number of cycles of stress, the category, type and location of material and the yield point of the base metal. The values required for a particular fatigue calculation can be found in the AASHO code. The constants for 500,000 cycles of stress are given in Table 3.2. These are the values which are used in the design program.

Table 3.2: Fatigue Constants

AASHO Category	Type and Location of Metal	Type of Maximum Stress	Equation	500,000 cycles		
				f_{ro}	α	k_2
A	Base Metal	Tension	3.6	20500	.78	.55
		Compr.	3.7	13300	.78	--
D	Metal Adjacent to Butt Weld	Tension	3.6	17200	.23	.62
		Compr.	3.7	10600	.23	--
F	Metal Adjacent to Fillet Weld	Tension	3.6	12000	0.0	1.0
		Compr.	3.6	12000	0.0	1.0
G	Weld Metal	Shear	3.6	10800	.36	.55

In the design of a rolled section, fatigue is a major factor in the cover plate design. The ends of the cover plates are determined by using fatigue calculations for the base metal adjacent to a fillet weld (Category F). The allowable stresses in this case are greatly reduced, thus requiring that the plates be cut off in regions of low moments. The remaining categories are checked in the design, but they usually do not govern the design of the girder.

3.3 Design Details

After the cover plates required for a particular rolled section are determined, the details of the design must be calculated. These design details include the determination of the weld sizes for the cover plates, the design of the required bearing stiffeners and the calculation of the shear connector spacing for a composite girder. In the optimization problem, each of these details must be completed for each design because they add a substantial cost to the girder.

The welds which must be determined for each cover plate include the end welds and the seal welds. The end welds are designed to carry the maximum flexural stress developed at a fixed distance from the ends of the plate. The code sets this distance as 1.5 times the width of the cover plate. The required stress is determined using either Equation 3.1 or 3.3. When the stress in the plate is known, Equation 3.8 is used to determine the force in the plate:

$$\text{Force} = f_{tot} \times A_{pl} \qquad 3.8$$

When the force in the plate is known, Blodgett [31] shows that the size of the required fillet weld is

$$t = \frac{\text{Force}}{.707\ L_w\ f_{all}} \qquad 3.9$$

In the above equation the L_w is the total length of the weld which, according to the AASHO code, is four times the plate width. The allowable weld metal stress (f_{all}) is the lesser value of the allowable static shear stress and the allowable fatigue stress, calculated using Equation 3.6 and the constants listed under category G from Table 3.2. The fillet weld equation then becomes

$$t = \frac{\text{Force}}{2.828\ W_{pl}\ f_{all}} \qquad 3.10$$

The required weld size is usually given in sixteenths of an inch. The code requires that a minimum size be used depending on the thicknesses of the plates being joined.

The design of the seal weld for a cover plate is basically similar to that of the end weld except that shear stress is used instead of flexural stress. Equation 3.4 is used to determine the maximum shear stress developed along the cover plate and then Equation 3.11 is used to determine the size of the seal weld:

$$t = \frac{1}{2}\ \frac{v}{(.707) f_{all}} = \frac{v}{1.414\ f_{all}} \qquad 3.11$$

The 2 in the denominator is present because there is a seal weld on each side of the cover plate. The code specifies that the seal weld must extend over the entire length of the plate in order to prevent corrosion and must have a minimum thickness of 5/16 in.

The bearing stiffeners are required to transfer the large web shear stresses, at the reactions, to prevent web crippling. In most of the wide-flange rolled sections, bearing stiffeners are not required. AASHO requires bearing stiffeners in those girders which have shear stresses in excess of 75 percent of the allowable shear stress. If stiffeners are required, they are designed as columns. The column section used is composed of the two stiffener plates and a portion of the web which is 18 times the web thickness in length. The fillet welds which connect the stiffener to the girder web are designed similar to the seal welds for the cover plates.

The shear connectors for a composite girder are designed to transfer the horizontal shear from the slab to the steel section. These shear connectors are placed transversely across the flange at either regular or variable spacings. The connectors are designed for fatigue as presented by Slutter and Fisher [33] and checked for ultimate strength. The spacing at any location can be found as follows:

$$\text{Spacing} = \frac{Z_r \, n_c}{S_r} \qquad 3.12$$

where S_r = the range of horizontal shear per inch. The allowable shear per connector depends on the number of fatigue cycles and the type and the size of the connector used. The horizontal shear per inch is found by using an equation similar to 3.4:

$$S_r = V_r \, Q/I \tag{3.13}$$

where V_r = the range of vertical shear due to live load plus impact.

The moment of inertia used in Equation 3.13 is the transformed moment of inertia in the positive region. If shear connectors are placed in the negative moment region, then the moment of inertia, for the negative region, is that of the steel girder and the longitudinal slab reinforcement.

In order to satisfy the ultimate strength requirements for the connectors, a specified number of connectors must be placed between the points of maximum moment and the points of zero dead-load moment. Equation 3.14 is used to determine the number of connectors required for the ultimate strength:

$$N = \frac{P}{.85\ S_u} \tag{3.14}$$

where P = the lesser value of the force in the steel girder and the concrete slab.

The ultimate strength of a shear connector depends on the size and the type of connector.

The final shear connector requirement which must be satisfied is an additional number of connectors to develop the slab stress around the points of contraflexure. The number of connectors required is

$$N_c = \frac{A_r \ f_r}{Z_r} \qquad 3.15$$

where A_r = the area of the steel reinforcement,

and f_r = the range of live load stress in the reinforcement (may be taken as 10,000 psi).

These additional connectors are needed only if the negative moment region is noncomposite.

3.4 Design Program

The design portion of the computer program operates on each wide-flange rolled section and completely designs the girder for the given problem. A complete design consists of the determination of the cover plates and their required welds, the design of the bearing stiffeners and the calculation of the shear connector spacing.

The design program is started by determining the lightest section which will carry the loads without cover plates. This section is completely determined and its cost becomes the base cost for the optimization problem. The program now enters a loop which designs all of the remaining rolled sections. A flowchart for this loop is shown in Figure 3.3.

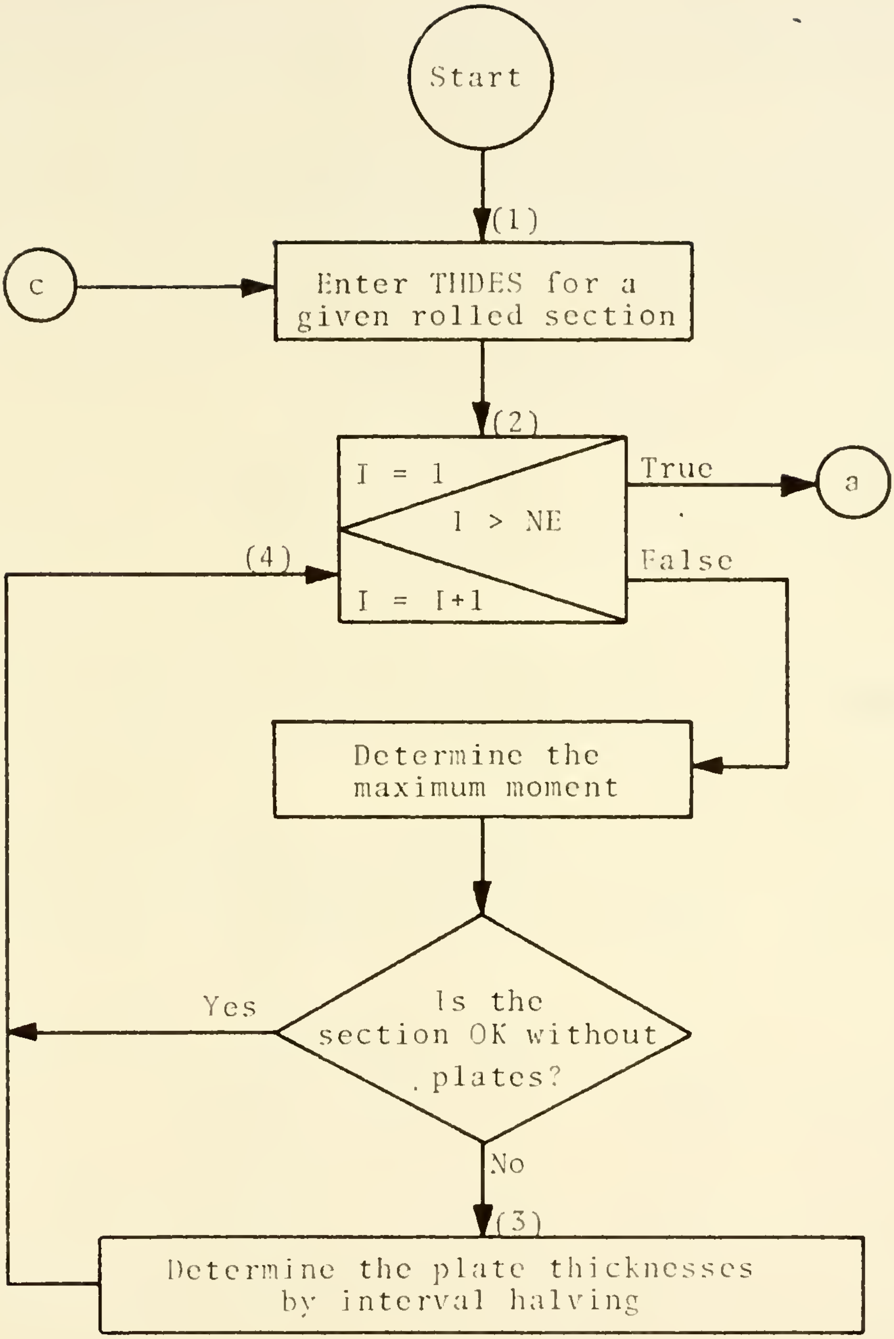

Start
(1)
c
Enter THDES for a given rolled section
(2)
I = 1
I > NE
I = I+1
True
a
False
(4)
Determine the maximum moment
Is the section OK without plates?
Yes
No
(3)
Determine the plate thicknesses by interval halving

Figure 3.3: Design Program Flowchart

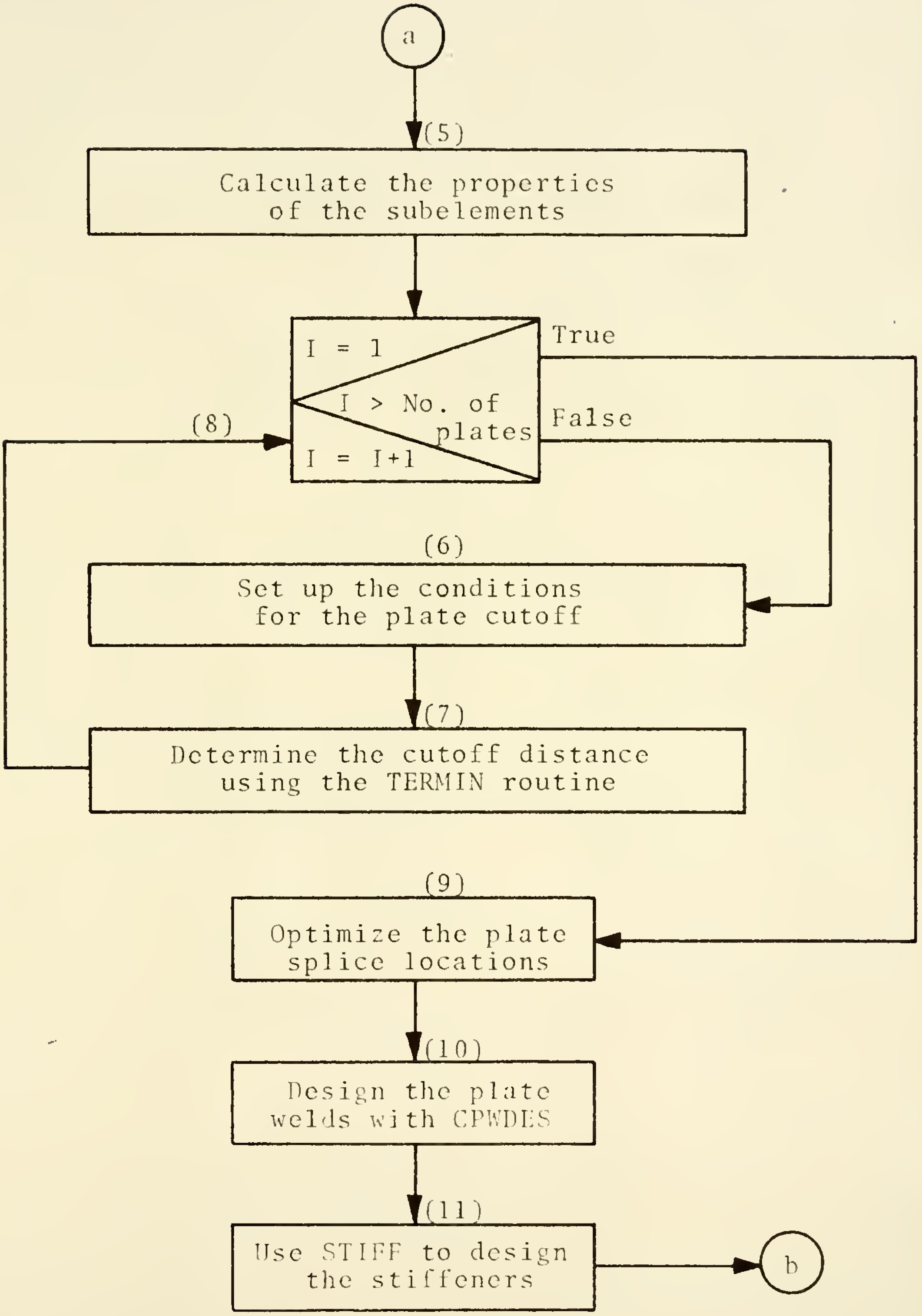

a
(5)
Calculate the properties of the subelements
I = 1
I > No. of plates
I = I+1
True
False
(8)
(6)
Set up the conditions for the plate cutoff
(7)
Determine the cutoff distance using the TERMIN routine
(9)
Optimize the plate splice locations
(10)
Design the plate welds with CPWDES
(11)
Use STIFF to design the stiffeners
b

Figure 3.3: (Cont'd.)

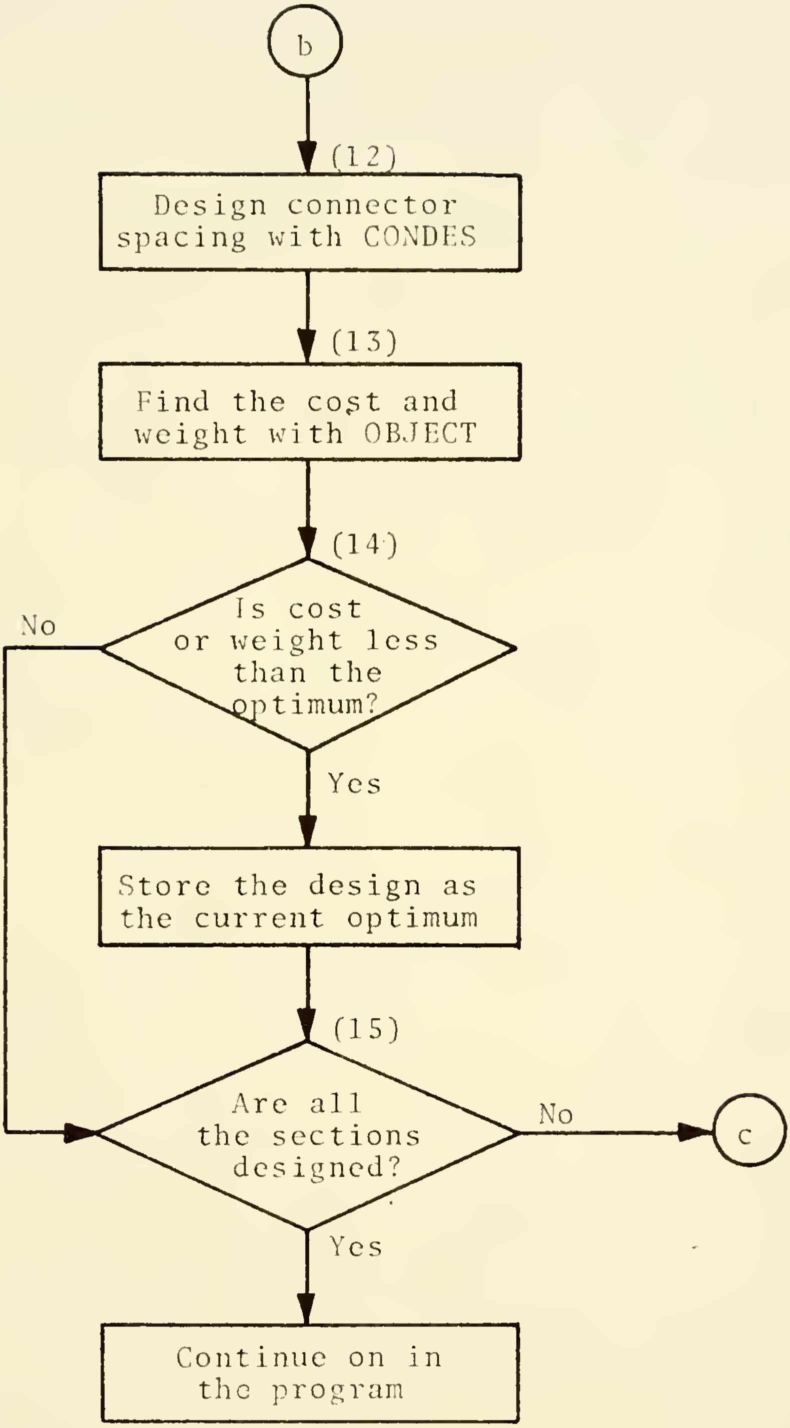

b
(12)
Design connector spacing with CONDES
(13)
Find the cost and weight with OBJECT
(14)
Is cost or weight less than the optimum?
No
Yes
Store the design as the current optimum
(15)
Are all the sections designed?
No
c
Yes
Continue on in the program

Figure 3.3: (Cont'd.)

Step 1

The rolled section being designed is transferred to the THDES subroutine where the thicknesses of the cover plates required for each subelement are determined.

Step 2

The THDES subroutine first determines the maximum moment in the subelement. With this moment determined, the rolled section is tried without plates. The individual stresses are determined and checked against the allowable static and fatigue base metal stresses. If these stresses are not violated, then no cover plates are required for the element.

Step 3

If a cover plate is required at a particular subelement, the width of the plate is determined based on the flange width of the rolled section. The thickness of the plate is determined by using interval halving. The interval halving technique uses only thicknesses of sixteenths of an inch. The thickness is varied until the allowable stress condition is satisfied. The thickness and width of the required plate are stored in the proper array.

Step 4

Steps 2 and 3 are repeated for each of the subelements along the girder. With all the plate sizes determined, the control is passed back to the main program.

Step 5

The properties of each subelement are calculated by using the SFMIC subroutine and these properties are transferred to the CPDES routine.

Step 6

The CPDES subroutine uses the fatigue theory discussed in Section 3.2 to determine the actual location of the ends of the required cover plates. The routine is used to determine all of the necessary conditions required to determine the cutoff point for each plate. The actual point is located by the TERMIN subroutine.

Step 7

The TERMIN subroutine uses an increasing step size optimization technique to determine the ends of the plates in the most efficient manner. The end of the plate is defined as the point along the girder where the stress does not violate the fatigue allowable stress. Once this point is determined, control is passed back to the CPDES routine.

Step 8

Steps 6 and 7 are repeated until the fatigue limits of each cover plate on the top and bottom of the girder are determined. With the limits found, the final cover plates for the top and bottom of the girder are determined in the FLOPT subroutine.

Step 9

The FLOPT routine uses the cost array and the cover plate arrangement to determine the optimum location of the cover plate splices. A dynamic programming method is used to accomplish this optimization. Upon completion of this optimization, the COVPL array is determined in its final form and control is returned to the main program.

Step 10

The CPWDES subroutine is now used to design all of the cover plate welds. The end welds for a particular plate are designed for both ends and the larger thickness is retained as the design. If the plate is butt welded to other plates at both ends, then the end weld thickness is set to zero. The seal welds are designed by the subroutine using Equation 3.11. After these calculations for each cover plate along the girder are completed, the control is returned to the main program.

Step 11

The STIFF routine is called to design the bearing stiffeners for each support. If the support does not require a stiffener, then the values of the BEAR array are set equal to zero. If a stiffener is required, the width and thickness are calculated such that the allowable bearing and the allowable compressive stresses are not violated. The welds which hold the stiffeners in place are calculated and the control is returned to the main program.

Step 12

The CONDES routine is used to determine the spacing of the shear connectors. The shear connector design is based on the use of 7/8-in. shear connector studs. The required spacing at each analysis point is determined and rounded off to the nearest lower multiple of three inches. The spacing is corrected for the ultimate strength discussed in Section 3.3 and the CONSP array is filled with the required spacing information. The routine sets the least number of spaces at a particular value at three. Upon completion of the spacing design, the control is returned to the main program.

Step 13

Once the spacing for the connectors is determined, the design of a rolled section is complete. The OBJECT subroutine is now called on to determine the weight and, if necessary, the cost of the given design.

Step 14

The cost or weight of the design is compared to the current optimum by the KEEP routine. If the girder is an improved design, the elements are stored in the proper arrays. If the girder is not an improved design, then the old optimum is retained.

Step 15

Steps 1 to 14 are repeated for each of the remaining rolled sections. After the last section is designed, the program takes the current optimum and continues into the remaining optimization phases of the program.

CHAPTER IV

PROBLEM OPTIMIZATION

4.1 Optimization Theory

Optimization, as stated by Beveridge and Schechter [34], "is the collective process of finding a set of conditions required to achieve the best result for a given situation." In other words, optimization is the process of obtaining a set of design parameters which either maximizes or minimizes a particular function. The function being optimized is referred to as the objective function. This function may be either mathematically simple or complex, depending on the nature of the problem. The actual form of this objective function usually determines the most appropriate method of optimization.

The design of a physical system ideally contains three steps. These steps, according to Wilde and Beightler [35], are:

1. Determination of the interaction of the system variables,
2. Development of a simple measure of effectiveness,
3. Determination, development, and solution of the most effective optimum-seeking method.

The first step is handled by developing a set of constraints for the particular problem being solved. These constraints can be imposed by design codes, tradition, or the designer. This step usually requires the greatest amount of effort.

The second step is the actual development of the objective function. The formulation of the function can range from being simple to being so complex it is almost impossible. The degree of difficulty in describing the function depends on the type of problem, the type of effectiveness being measured, and the degree of accuracy desired. The less the degree of accuracy of the objective function, the less accurate will be the developed optimum solution.

The third step leads the designer to many different types of optimization procedures. The advantages of each method vary with the problem being solved. It is the designer's responsibility to know the limitations, the advantages and the disadvantages of each method as they relate to the particular problem being solved.

In the design program for the optimization of a rolled-section highway bridge girder, there are actually two different optimum-seeking methods employed. The method of exhaustive search is used as the basic design method and dynamic programming is used to determine the locations of the cover plate splices. The thicknesses of the cover plates are determined by using interval halving to solve the governing equation.

The exhaustive search technique, sometimes called exhaustive enumeration or brute force method, simply evaluates the objective function for each of the problem possibilities and picks the optimum solution directly. This method can be very tedious if the number of design possibilities is large. It is therefore used only when the overall design space for the problem is fairly limited in size.

Since the wide-flange sections used for highway girders include only the 36- and 33-in. sections, there are only 18 sections to be considered. The value of the objective function is determined for each wide-flange section after it has been completely designed, including the cover plates, the bearing stiffeners and the shear connectors. The design with the minimum value of the objective function is then chosen as the optimum design.

By setting up the wide-flange sections in descending weight order, the number of executions in the exhaustive search can be reduced. Starting with the heaviest, and proceeding in the descending weight order, each section is tried without cover plates at the critical moment location, until the lightest girder which is acceptable is determined. By assuming that the cost of the shear connectors and stiffeners for a uniform section are independent of the section used, all of the sections heavier than the one which has been determined can be eliminated from the search. This

means that a number of the rolled sections need not be designed completely and a resulting savings in computation costs is realized.

Interval halving is a technique which can be used to solve equations when the simple, direct methods are inappropriate. The method uses a bounded interval which is continuously reduced in size until a certain tolerance is obtained. By reducing the interval size by a factor of one-half with each calculation, the total number of necessary calculations is greatly reduced and there is a resulting savings in calculating costs. The two necessary conditions for the interval halving technique are:

1. The function must be continuously increasing or decreasing.
2. The initial boundaries must be specified.

The first condition enables the method to completely eliminate one half of the interval with each calculation, and the second condition enables the first calculation value to be specified.

The method begins with the specified interval and calculates all the necessary information for the interval midpoint. If all of the constraints are satisfied for this midpoint value, then the value becomes the new upper boundary. This is possible because the function is continuously increasing or decreasing and the higher values of the variables will also satisfy the constraints, but will cause a

larger cost. If the constraints are not satisfied, then the midpoint value becomes the lower boundary. The new interval is used in the same manner as the initial interval. The entire procedure is repeated until the remaining interval is less than the tolerance desired. Either the upper or the lower bound can then be used as the optimum solution. In most problems the average of the two bounds is considered as the solution, provided that the final interval is small enough.

In determining the cover plate thicknesses for each subelement, there are many calculations required, including the section properties for each plate size, the bending stress at the point in question and the allowable stresses for the fatigue and static conditions. Since, for each moment condition and rolled section there is only one plate size which will give a stress equal to the allowable stress, the function for the thickness can be considered to be continuously decreasing. The initial boundary conditions can be specified by using the requirements of the AASHO code. With the two conditions for interval halving satisfied, the method can be used for the determination of the cover plate thicknesses.

In order to use interval halving, however, the method must be adjusted so that the thicknesses will be determined in values of sixteenth-inch increments. This is accomplished by adjusting the initial interval size and by making the

boundaries integers. Each halving process then produces an integer which corresponds to the number of sixteenths of an inch in the plate thickness. The calculations are completed for the thickness and the interval is reduced. The process is continued until the final interval size is 1.0. The optimum plate size is then stored in the proper array location.

Dynamic programming is a method used in sequential system optimization problems. It can be applied to situations in which many decisions are required, as long as the decisions made at later stages do not affect the performance of earlier stages. A mathematical formulation of the dynamic programming method can be found in the optimization texts by Wilde and Beightler [35], Pierre [36] or Denn [37]. All of the mathematical formulations are based on Bellman's [38] principle of optimality.

Dynamic programming is a method of decomposition which divides the given problem into a set number of individual problems. In determining the optimum condition for section i, the only quantities which are used are the values of the variables in sections i+1 and i, and the value of the objective function for the system up to and including section i-1. The objective function is determined for all of the possible combinations of the variables in section i and i+1. The values of the objective function are then added to the value of the function for the problem through

section i-1. The optimum variable can then be directly chosen and placed in an appropriate array location defined by the section number i and the variable value in section i+1.

The method begins by calculating the variable combinations for all of the sections and their adjacent sections in the problem sequence. When the last section is reached, the final values of the objective function are calculated. By comparing these values, the optimum for the last section is determined. Then, by working backwards, all of the variables in each section can be determined. This process leads to an optimization of the entire problem.

The dynamic programming method is used to determine the final cover plate combination for a given rolled section. The actual problem is decomposed into sections corresponding to the cover plates which have been defined by the fatigue stress conditions. The variable of each section is the plate thickness, and the function being minimized is the total cost of the cover plate. By varying the thickness of each section and determining the total cost of the plate by the procedure described above, the optimum location of the cover plate splices can be located. This method is similar to the flange smoothing method used by Goble and Razani [17]. DeSantis and Goble [18] illustrate this method for the optimization of the flanges for a welded plate girder.

4.2 Objective Function

The term objective function refers to the means by which the program evaluates the merit of a particular design. For the rolled-section highway girder design, the function can be either the total weight or the total cost of the girder. The designer specifies which method of optimization is desired by the value used for IPT6. If IPT6 has a value of 0, then the optimization is based on cost. The equation for the total girder cost is

Total Cost = Cost of the rolled section +
Cost of the cover plates +
Cost of the shear connectors + 4.1
Cost of the bearing stiffeners +
Cost of the welding.

If the value of IPT6 is 1, then the optimization is based on the weight of the girder. The equation for the total girder weight is

Total Weight = Weight of the rolled section +
Weight of the cover plates +
Weight of the bearing stiffeners + 4.2
Weight of the shear connectors.

The weight function first determines the volume of steel used in each of the individual portions of Equation 4.2. A unit weight of 490 pounds per cubic foot is used to convert

the volumes into weights. These weights are added together to get the total weight.

If the cost of a girder is desired, then the weights are calculated as above and then converted into costs. The costs of the rolled section, the cover plates, and the bearing stiffeners are determined by multiplying each of the weights by their respective unit costs. These unit costs are expressed in dollars per pound.

The cost of the shear connectors consists of two separate elements. The first element is the material cost of the connectors and is based on the total weight of the connectors. The cost is calculated by multiplying the total connector weight by the unit cost. This unit cost is also expressed in dollars per pound. The second element is the installation cost and is based on the total number of connectors. This cost is calculated by multiplying the total number of connectors by the installation cost per connector.

The welding costs are determined for all the welds on the girder. The welds which are considered are the cover plate butt welds, the cover plate seal welds, and the welds used to connect the stiffeners to the girder. The weld cost can be found for any weld using the following equation:

$$C_{weld} = C_1 + C_2(L) + C_3(V) \qquad 4.3$$

where C_1 is the fixed cost of the weld, $C_2(L)$ is the variable cost which is a function of the weld length and $C_3(V)$ is the variable cost which is a function of the weld volume.

The fixed cost of a weld accounts for all of the costs involved in the set-up time, the inspection, and the other fixed items of cost. The variable cost, based on the weld length, reflects the costs of x-ray inspection and joint preparation.

The variable cost, based on volume, reflects the actual cost of the weld material. The volume of a weld depends on the size and shape of the weld. For a fillet weld, the volume is based on a weld cross section which is triangular. For a butt weld, the volume is based on the cross section of a double vee butt weld. Each of these welds is the standard weld used in highway girder design.

The objective function for this problem is similar to the one found in the GAD-1 program developed by Goble and DeSantis [19]. This is done so that final comparisons between rolled-section girders and welded plate girders can be made. These comparisons can only be accurate if the objective functions for both types of design are similar in form and content.

4.3 Constraints

The problem constraints are the limitations placed on various parts of the design in order to limit the total design space. There are two types of constraints in an

optimization problem. One type of constraint, referred to as a side constraint, is imposed on the design by functional limitations. The second type, called a behavioral constraint, is imparted to the design by a specific specification.

A functional, or side, constraint used in the girder design problem is the total number of rolled sections designed. Since the design of a highway bridge girder is usually limited to the 36- and 33-in. wide-flange sections, the limited number of design possibilities actually becomes a side constraint for the problem.

The allowable width of the cover plates is also controlled by a functional constraint. The plates are limited to sizes of whole inches and must be smaller than the width of the rolled-section flange. The reason for this is to make the welding process more efficient. With this constraint, the widths of the cover plates are limited to 10, 14, or 15 in.

The behavioral constraints for the highway girder design are specified in the American Association of State Highway Officials bridge design code [20]. The AASHO code specifies all of the design criteria, which must be converted into the problem constraints, for the design of a rolled-section girder. These behavioral constraints are divided into the following four groups:

1. Static stress constraints,
2. Fatigue stress constraints,

3. Cover plate size constraints,
4. Secondary design constraints.

The static stress constraints are based on the required allowable stresses with no regard to fatigue considerations. There are actually five different static stress conditions which must be satisfied before a design can be considered acceptable. Four of the conditions consider the bending stresses in the girder and the fifth considers the shear stress in the girder web. The five constraints are:

1. The allowable flexural tension stress in the extreme fibers of rolled shapes or built-up sections is 0.55 F_y where F_y is the yield point stress.
2. The allowable flexural compression stress in the extreme fibers of rolled shapes and built-up sections is 0.55 F_y when the compression flange is continuously supported by being embedded in concrete.
3. The allowable flexural compression stress in the extreme fibers of rolled shapes or built-up sections is determined by Equation 4.4:

$$F_b = 0.55\,F_y\left[1.0 - \frac{3F_y}{\pi^2 E}\left(\frac{\ell}{b}\right)^2\right] \qquad 4.4$$

   This constraint reduces the allowable compressive stress in order to prevent the buckling of the compressive flange.

4. The allowable compression stress in the extreme fibers of the concrete slab is $0.4\ f'_c$ where f'_c is the 28-day concrete strength.
5. The allowable shear stress in the girder web is $0.33\ F_y$.

The fatigue stress constraints reduce the allowable stresses because of the fluctuations in stress caused by the repeated applications of the live load. The fatigue stress conditions are governed by Equations 3.6 and 3.7. The fatigue theory used in the design program is described in Section 3.2.

The maximum permissible thickness of a cover plate is governed by Equation 4.5:

$$t_{cp_{max}} = 1.5\ t_f \qquad 4.5$$

Equation 4.6 governs the length of the cover plate:

$$L_{cp_{min}} = 2\ D + 3 \qquad 4.6$$

where D is the depth of the rolled section in feet.

Equations 4.5 and 4.6 set the requirements for a cover plate on a given rolled section and make up the cover plate size constraints.

The final group of design constraints control the secondary details of the girder design. The details which

are controlled by this group are the shear connector spacings, the bearing stiffener design, and the weld design.

If the design is composite, then the following four constraints control the spacing of the shear connectors:

1. The shear connectors are spaced so that the horizontal shear is transferred from the slab to the steel girder. This spacing is designed according to fatigue stress conditions.
2. The design of the shear connector spacing is checked for ultimate strength.
3. Extra connectors must be provided at the points of contraflexure.
4. The maximum permissible spacing is 24 in.

The equations and the theory used in the design of these shear connector constraints can be found in Section 3.3.

The design of the bearing stiffeners is controlled by the following constraints:

1. Stiffeners are required at a support if the shear stress in the web at that support is greater than 75 percent of the allowable shear for girder webs.
2. The minimum thickness for a stiffener is given by

$$t_{min} = \frac{b'}{12}\sqrt{\frac{F_y}{33000}} \qquad 4.7$$

where b' is the width of the stiffener.

3. The bearing stress between the stiffener and the rolled-section flanges must be less than 0.80 F_y.

4. The compressive stress in the stiffener must be less than the allowable column stress determined from Equation 4.8:

$$F_a = \frac{0.55F_y}{1.25}\left[1.0 - \frac{\left(\frac{0.75L'}{r}\right)^2 F_y}{4\pi^2 E}\right] \qquad 4.8$$

where L'/r is the slenderness ratio of the stiffener and a section of the web which is $18t_w$ inches wide.

The final secondary design constraint is used to control the weld design. The allowable static shear stress in the weld material (F_V) is specified by Equation 4.9:

$$\begin{aligned} &\text{If } F_y \leq 36 \text{ ksi}, \quad F_v = 12.4 \text{ ksi.} \\ &\text{If } F_y > 36 \text{ ksi}, \quad F_v = 14.7 \text{ ksi.} \end{aligned} \qquad 4.9$$

The allowable shear stress for the weld material must be reduced for fatigue by using Equation 3.6 and the constants under category G in Table 3.2.

CHAPTER V

COMPUTER PROGRAM

5.1 Description of the MAIN Program

The computer program for the design of rolled-section girders consists of two parts. The analysis portion is used to determine the design conditions for the problem and the design portion is used to determine the optimum combination of rolled section and cover plates. The entire program is composed of a MAIN program and thirty-four subroutines.

The purpose of the MAIN program is to call on the subroutines in the correct sequence to produce the optimum design for the particular girder. The following description is a step-by-step summary of the MAIN program. A flowchart for this program is presented in Figure 5.1.

Step 1

The rolled section information required for the design program is read from data cards. All the information for the eighteen 36- and 33-in. wide-flange sections is already on cards supplied with the program.

Step 2

The READIN subroutine is called to read all of the data for the design problem. A detailed description of READIN is presented in Appendix C.

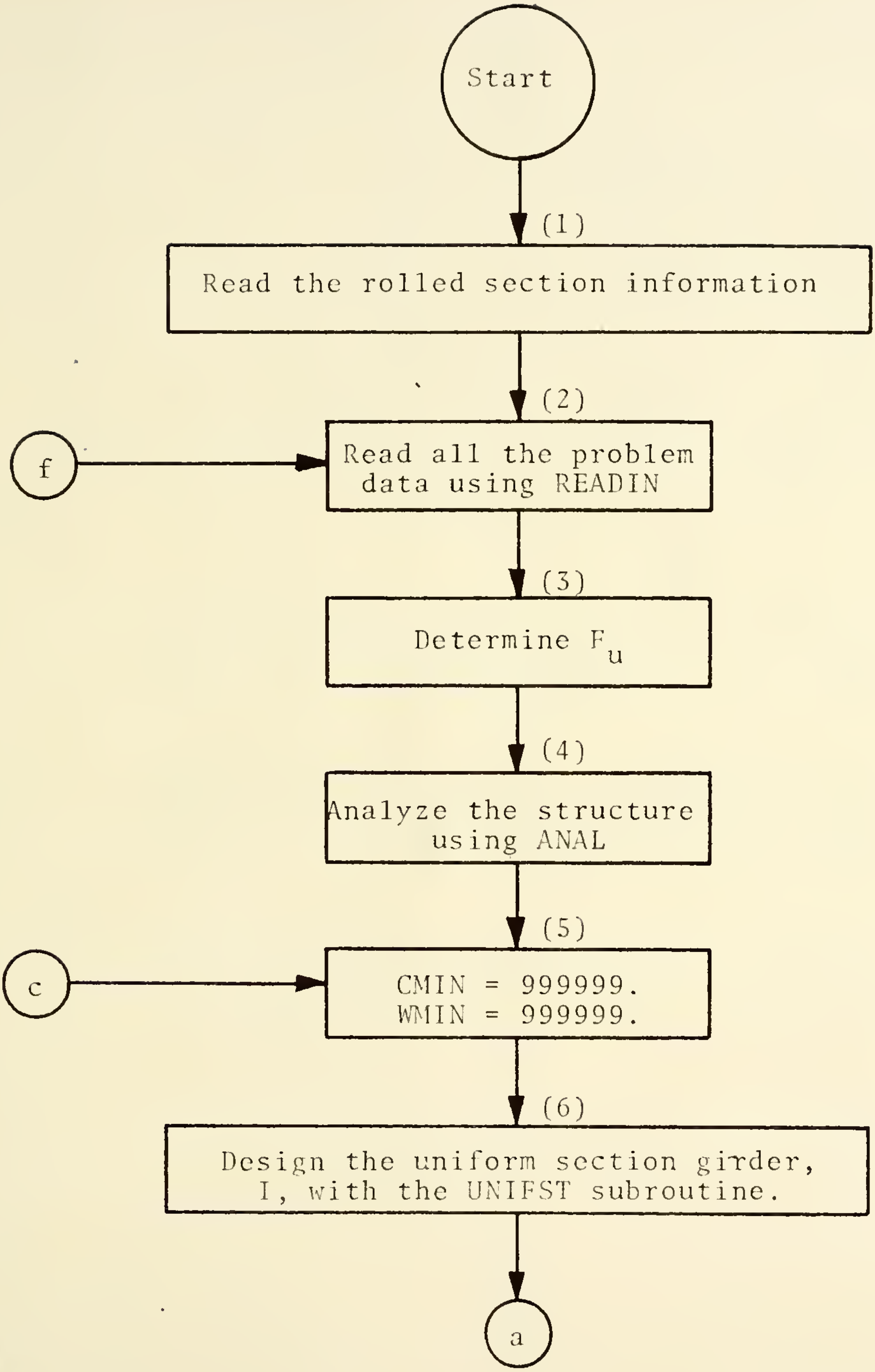

Start
(1)
Read the rolled section information
(2)
f
Read all the problem data using READIN
(3)
Determine F_u
(4)
Analyze the structure using ANAL
(5)
c
CMIN = 999999.
WMIN = 999999.
(6)
Design the uniform section girder, I, with the UNIFST subroutine.
a

Figure 5.1: Flowchart for the MAIN Program

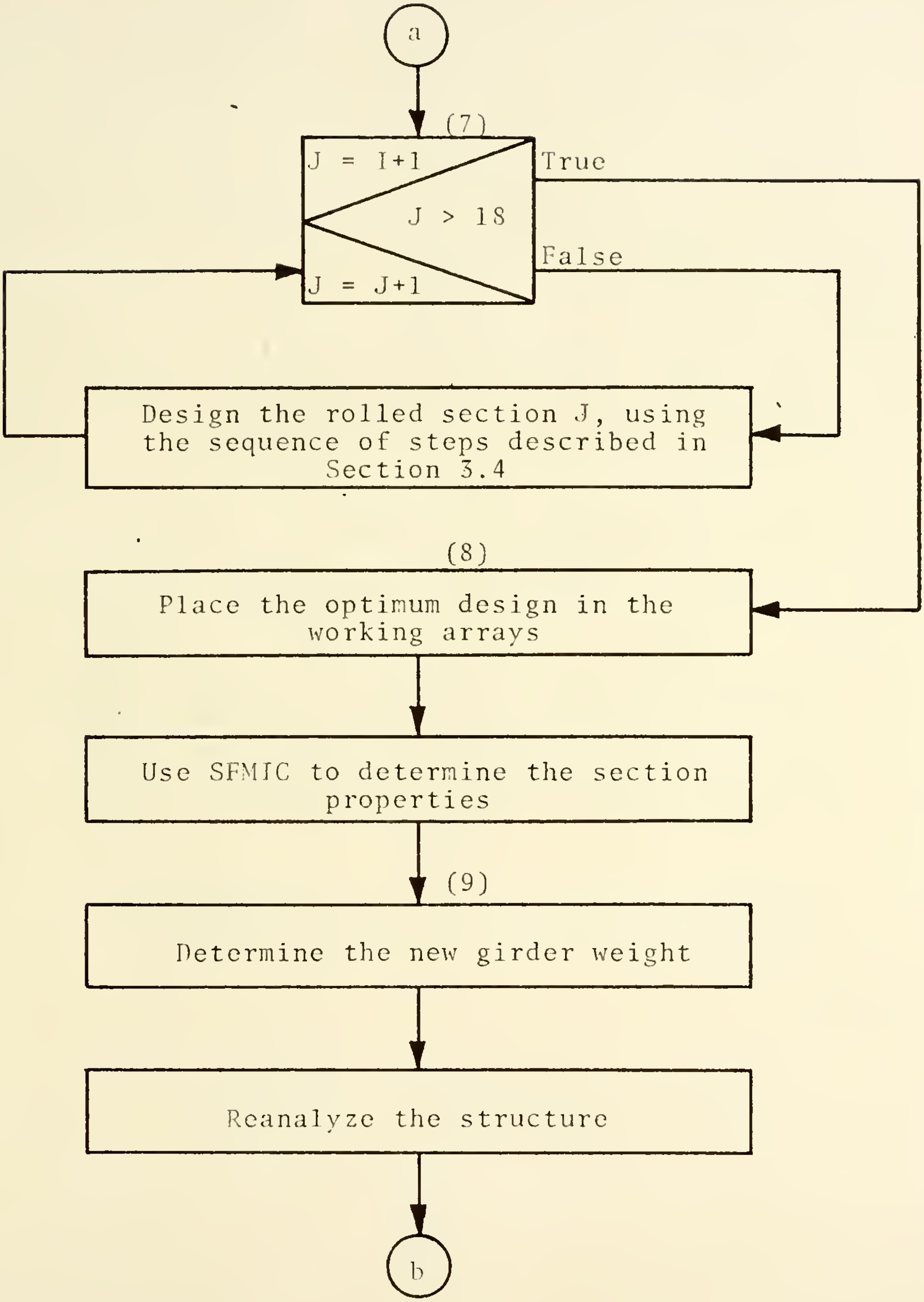

a
(7)
J = I+1
J > 18
J = J+1
True
False
Design the rolled section J, using the sequence of steps described in Section 3.4
(8)
Place the optimum design in the working arrays
Use SFMIC to determine the section properties
(9)
Determine the new girder weight
Reanalyze the structure
b

Figure 5.1: (Cont'd.)

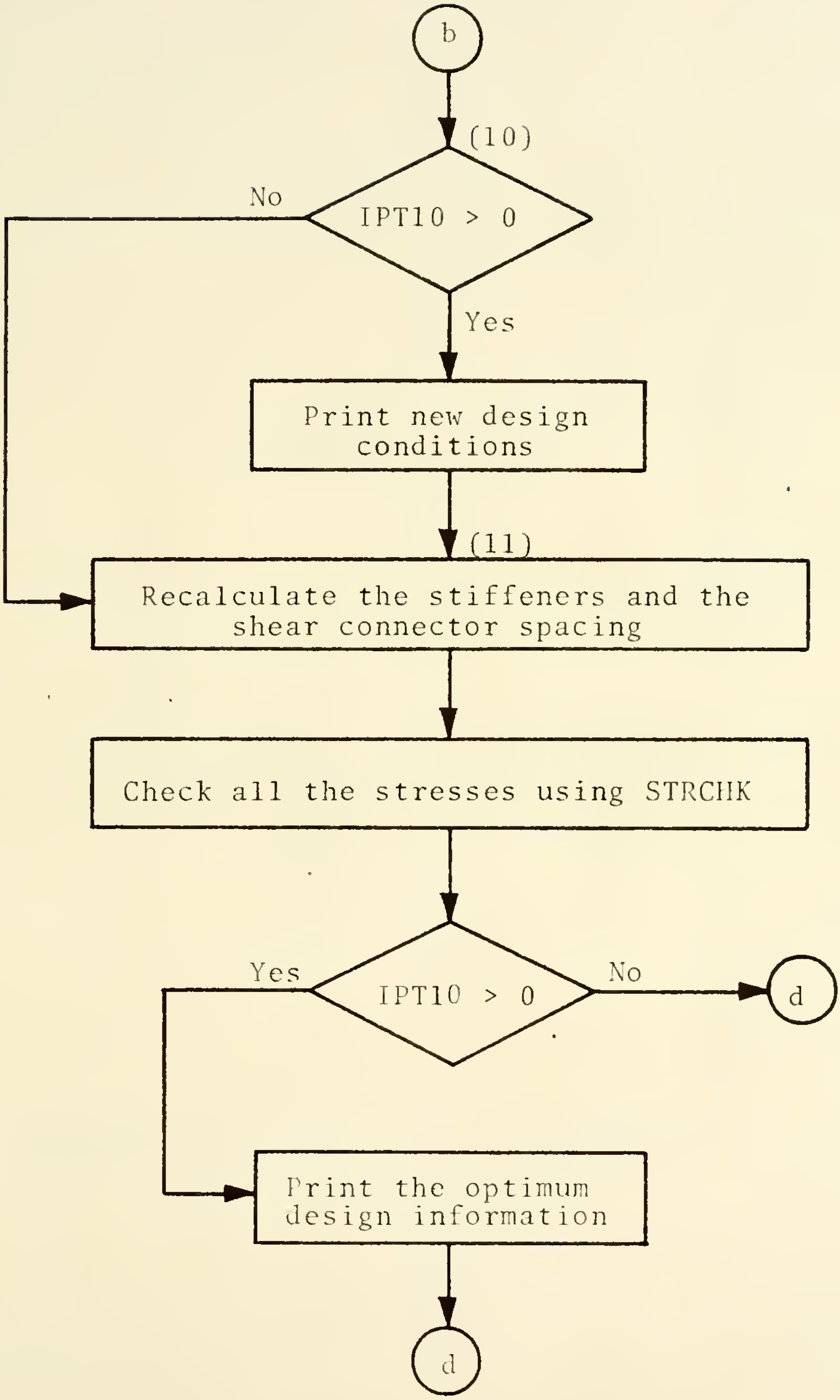

b
(10)
IPT10 > 0
No
Yes
Print new design conditions
(11)
Recalculate the stiffeners and the shear connector spacing
Check all the stresses using STRCHK
IPT10 > 0
Yes
No
d
Print the optimum design information
d

Figure 5.1: (Cont'd.)

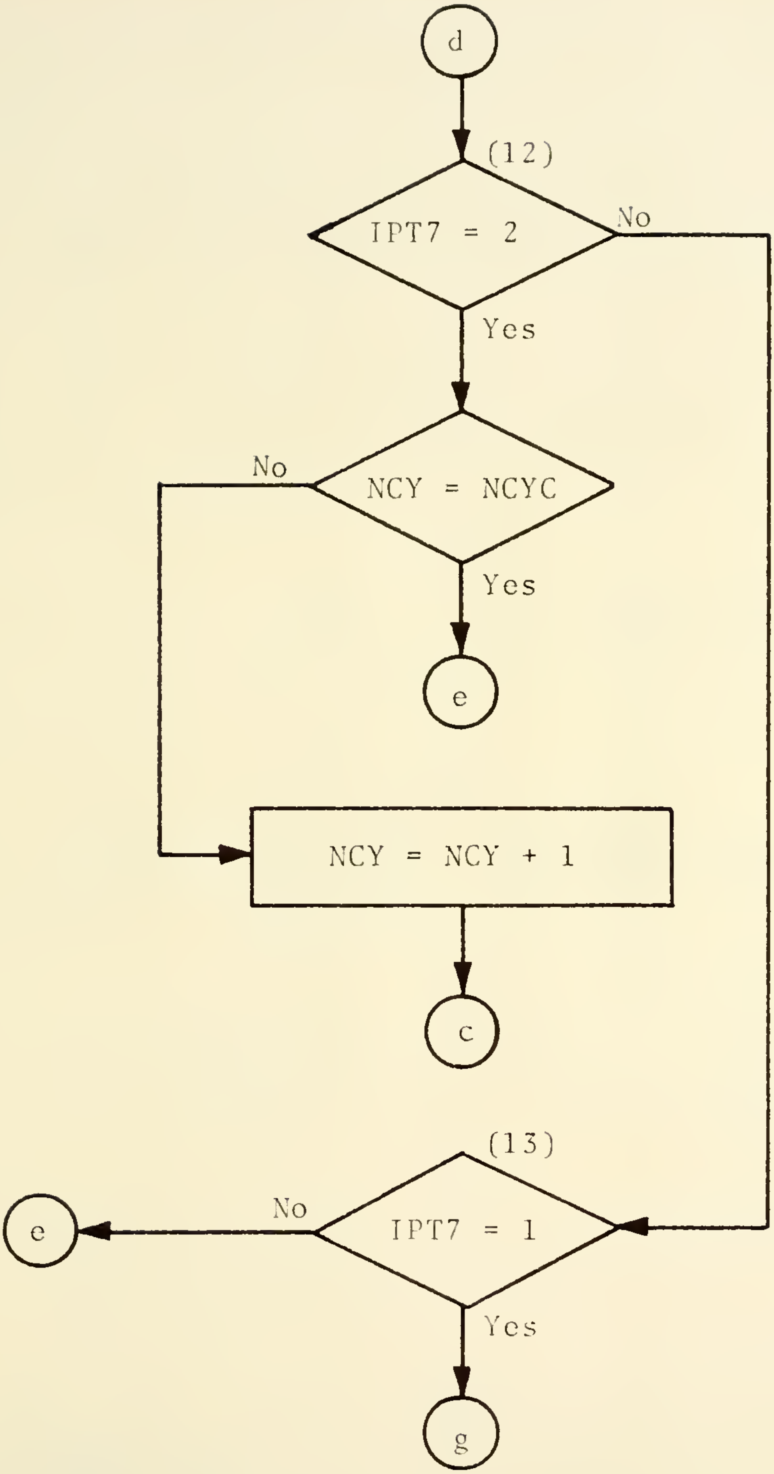


Figure 5.1: (Cont'd.)

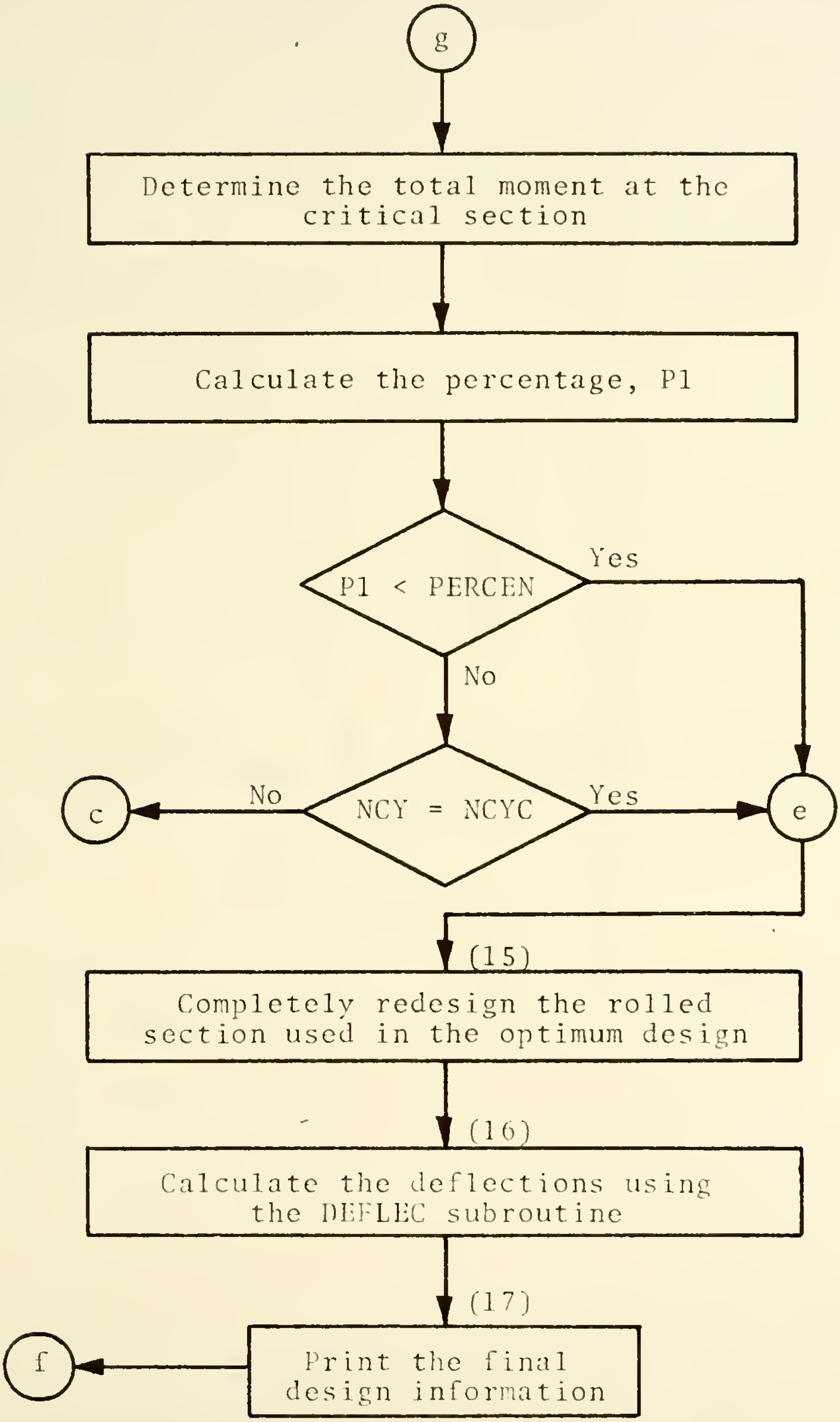

g
Determine the total moment at the critical section
Calculate the percentage, P1
P1 < PERCEN
Yes
No
NCY = NCYC
No
Yes
c
e
(15)
Completely redesign the rolled section used in the optimum design
(16)
Calculate the deflections using the DEFLEC subroutine
(17)
Print the final design information
f

Figure 5.1: (Cont'd.)

Step 3

The minimum tensile strength used in the fatigue calculations is determined from the yield point stress of the material used in the problem. This tensile strength is a measure of the minimum ultimate strength of the steel.

Step 4

The ANAL subroutine is called to analyze the given problem and to determine all of the design reactions, the design moments, and the design shears. These conditions are used to determine the optimum girder for the problem.

Step 5

The minimum weight and the minimum cost variables are initialized at very high values.

Step 6

The UNIFST subroutine is used to design the uniform section girder. This girder is the lightest rolled section which satisfies all of the design constraints without the use of cover plates. This section then becomes the initial design.

Step 7

A design loop, which completely determines the girders composed of the sections lighter than the one used as a uniform section, is begun. This loop is described in Section 3.4 and a flowchart is presented in Figure 3.3.

Step 8

After the design loop is completed, the optimum design for the cycle is stored in the optimum storage arrays. This design is then placed into the working arrays of the program and all of the section properties are computed using the SFMIC subroutine.

Step 9

The new unit dead weight of the girder is determined and the problem is analyzed again, using the new distribution of the moments of inertia. If the problem is a continuous girder, then the ANAL subroutine is called to do the complete reanalysis. If the problem is a simple-span girder, the dead-load design conditions are changed directly in the MAIN program. This is the only change required because the moments in a simple-span girder are independent of the cross-section properties.

Step 10

If IPT10 is greater than 0, all of the design conditions are printed out.

Step 11

Using the new design conditions, the stiffeners and connector spacing for the optimum design are recalculated. All of the stresses in the girder are determined by the STRCHK subroutine and a new value of the objective function

is determined. All of the information regarding this design is then printed if IPT10 is greater than 0.

Step 12

If the design cycle parameter, IPT7, equals 2 and the maximum number of cycles has not been reached, the counter is increased by one and control is returned to step 5. Steps 5 thru 11 are repeated until the specified maximum number of cycles is reached. Control is now passed to step 15.

Step 13

If IPT7 equals 1, the total moment at the critical section is determined and the percent change from the previous analysis is computed. If this percentage is less than the specified percentage, control is passed to step 15. Otherwise, control is returned to step 5, where steps 5 thru 11 are repeated until the percentage is less than that specified.

Step 14

If IPT7 equals 0, no redesign cycles are required and control is passed to step 15.

Step 15

Using the final conditions, the rolled section used in the optimum design is completely recalculated using the sequence of steps described in Section 3.4.

Step 16

The required deflections are calculated, using the moment of inertia distribution of the final design. These deflections are calculated by the DEFLEC subroutine.

Step 17

All of the information pertaining to the final design is printed in the output. This includes the section properties, the required cover plates, the required stiffeners, the shear connector spacing and all of the deflections. The cost or weight of the design is also presented, depending on the value of IPT6.

Step 18

Control is returned to Step 2 where the program is either terminated or the data for a new design problem are read.

5.2 Sample Problems

In order to show the usefulness of the computer program, two sample girders have been designed. Both problems are presented in the U. S. Steel Manual [31]. The costs of the manual design and program design are compared below.

The first problem is a two-span girder having equal span lengths of 70 ft. The design is based on a composite section having the properties shown in Table 5.1. The design loading is the AASHO HS20 loading.

Table 5.1 Composite Section Properties

Property	Value
SLABWD	84.00 in
SLABTH	7.00 in
FPC	3000.00 psi
MR	8.00
AS	4.34 sq in
HAUNCH	1.875 in

Table 5.2 Unit Costs

Item	unit	Cost in Dollars
Rolled Section	lb	0.15
Cover Plates	lb	0.12
Stiffeners	lb	0.12
Connectors	lb	0.00
Connector Installation	ea	0.75
Fixed Weld Cost	ea	20.00
Weld Material Cost	cu in	2.00
Weld Length Cost	in	0.0

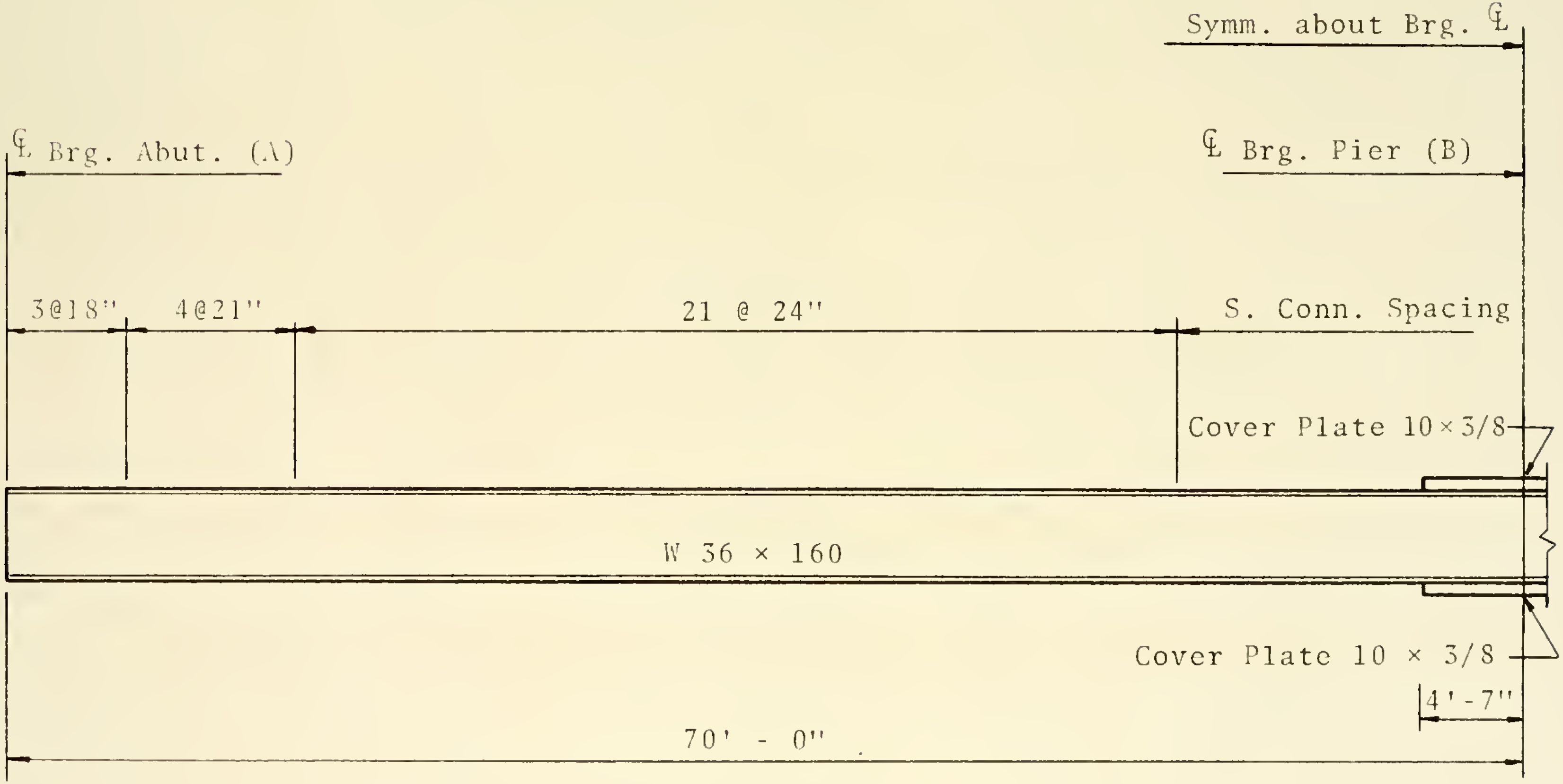
The optimization of the girder is based on the total cost. The unit costs used in the optimization procedure are shown in Table 5.2

The final design produced by the computer program is shown in Figure 5.2. The itemized costs and weights of both the computer design and the manual design are given in Table 5.3.

Table 5.3: Costs and Weights for Problem No. 1

Item	U. S. Steel Manual		Computer Program	
	Weight	Cost	Weight	Cost
Rolled Section	18900.0	$2835.00	22400.0	$3360.00
Cover Plates	2361.6	$283.40	243.1	$28.16
Stiffeners	---	---	---	---
Connectors	202.2	$193.50	181.0	$174.00
Welds	---	$628.29	---	$206.90
Totals	21463.8	$3940.19	22824.1	$3769.06

Table 5.3 shows that the optimization program produced a girder which is approximately 6.0 percent heavier than the girder found in the manual. The optimum design, however, costs 5.0 percent less than the manual design. Most of the cost reduction can be found in the cover plate cost. The total cost of the plates, including both material and fabrication costs, for the optimum girder is $235.06. The



Notes: Use 4 connectors in each row.
Use 5/16" end welds.
All seal welds are 5/16".

Figure 5.2: Optimum Design for Problem No. 1

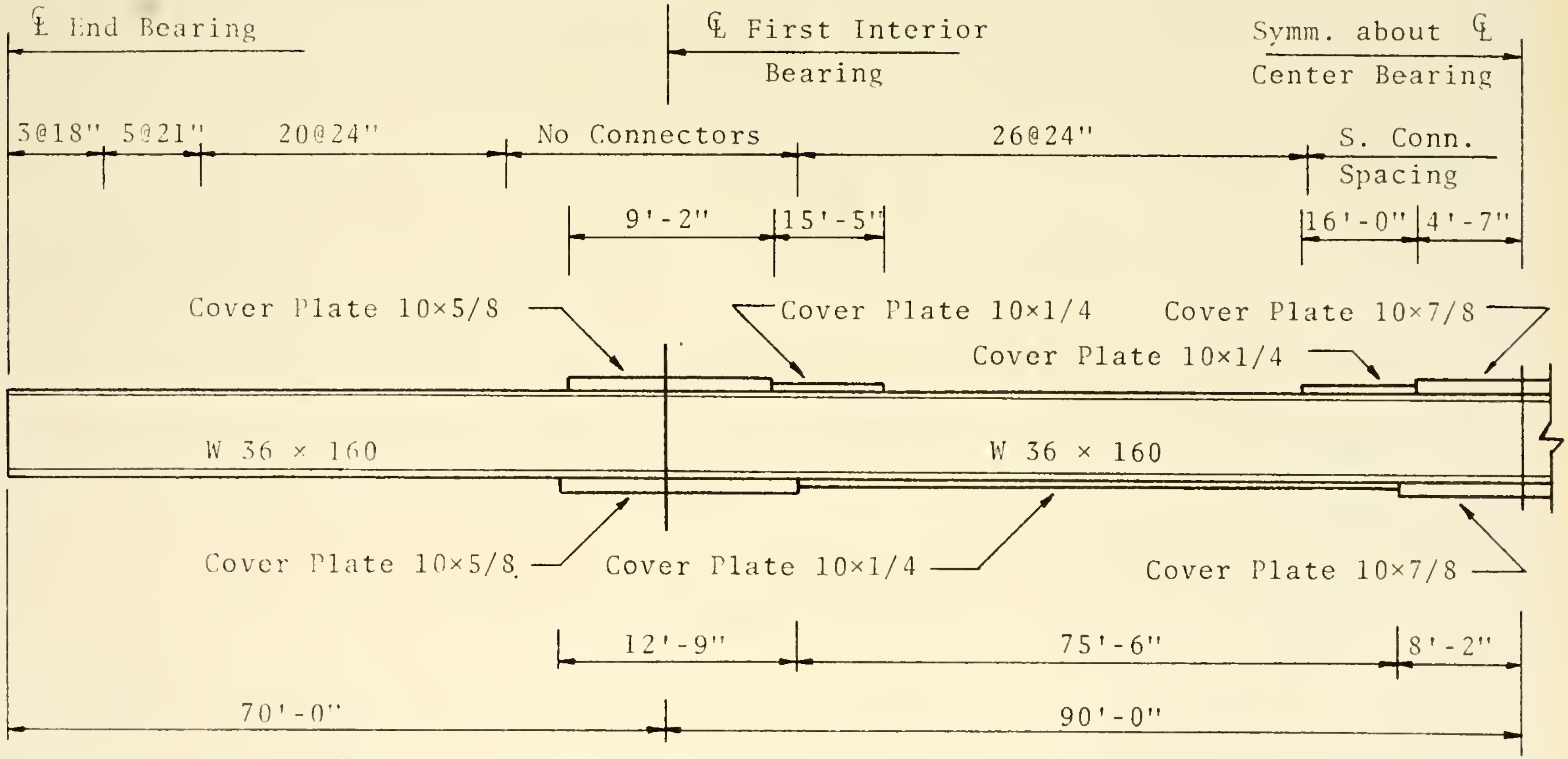
same costs for the manual design give a total of $911.69 or approximately four times as much as the program solution. This cost difference offsets the increased cost of the rolled section used in the optimum design.

The second example is a four-span composite girder. The outer spans are 70 ft and the interior spans are 90 ft. The composite section properties, the loading conditions and the cost parameters are the same as for the first problem. The program design and the itemized costs and weights are shown in Figure 5.3 and Table 5.4 respectively.

Table 5.4: Costs and Weights for Problem No. 2

Item	U. S. Steel Manual		Computer Program	
	Weight	Cost	Weight	Cost
Rolled Section	43200.0	$6480.00	51200.0	$7670.00
Cover Plate	9855.0	$1182.60	3509.6	$420.93
Stiffeners	---	---	---	---
Connectors	475.0	$458.00	348.0	$336.50
Welds	---	$1598.44	---	$1251.68
Totals	53520.0	$9719.04	55057.6	$9679.11

Table 5.4 shows that the computer solution is approximately 2.9 percent heavier and only 0.5 percent cheaper than the manual design. These figures show that there is a slight saving in the computer solution. The small amount



Notes: Use 4 connectors in each row.
All seal welds are 5/16".
Use 5/16" and 1/4" end welds.

Figure 5.3: Optimum Design for Problem No. 2

indicates that the manual solution approaches, for these cost figures, the optimum design. The similar costs show that the computer not only approaches the optimum solution, but also produces a practical design.

The computer program is approximately 4000 cards in length and was developed using the CDC 6500 computer. The compilation time for the program is about 48 seconds. The execution times for the two sample problems were 60 and 40 seconds respectively.

The 10 × 1/4 plates, shown in Figure 5.3, violate width-to-thickness ratios of the AASHO code. The program is being modified such that the cover plates conform to the code requirements.

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

The cost optimization of a highway bridge girder, composed of a rolled section acting compositely with the bridge deck, has been formulated and programmed. An objective function, based on the material and the fabrication costs, has been developed and used in the examples. The design of the girder is controlled by the 1969 AASHO code.

Due to the complex loadings required on a highway bridge, the method of influence lines is used in the analysis. The analysis portion of the program produces the design conditions for the problem, which are used to design and optimize the required girder.

Since the number of possible designs is fairly limited, the program uses the exhaustive search technique. The techniques of interval halving and dynamic programming are also used within the design program. Interval halving is used to determine the thicknesses of the cover plates for the subelements and dynamic programming is used to determine the optimum location of the cover plate splices.

The computer program developed will solve a simple-span problem as well as a continuous problem of up to four spans. The girder can be either composite or noncomposite and is designed for static and fatigue loads. All the design details are developed and then listed in the output. No attempt is made to design the girder for deflection, but the total deflections for the final design are tabulated. The engineer has the option to optimize the design based on either the total weight or the total cost.

6.2 Recommendations for Further Research

1. Beams with different steel strengths may be considered.
2. Varying the rolled section at points of zero dead-load moment may be considered.
3. The effect of field splices on the optimization problem could be established.
4. A more accurate objective function could be developed, provided that cost information could be obtained from steel fabricators.
5. A complete optimization study of rolled-section girders and plate girders could be made using the program in conjunction with GAD-I.
6. The effect of the girder spacing on the optimum solution could be determined.

7. The effect of composite action in the negative moment region on the optimum solution could be studied.
8. The effect of a variable slab thickness on the problem optimization could be determined.
9. An optimization of the bridge deck and stringer portion of a highway bridge could be developed.
10. An optimization study of the various types of highway overpass structures, such as plate girders, rolled-section girders, reinforced and prestressed concrete girders and concrete slab bridges, could be made with the ultimate outlook on optimizing the entire bridge design project.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Drucker, D. C. and R. T. Shield: "Bounds in Minimum Weight Design," *Quarterly of Applied Mathematics*, 1957.

2. Drucker, D. C. and R. T. Shield: "Design for Minimum Weight," *Proceedings on the 9th International Congress of Applied Mechanics*, Brussels, 1956.

3. Faulkes, J.: "The Minimum Weight Design of Structural Frames," *Proceedings of the Royal Society*, London, 1954.

4. Heyman, J.: "On the Absolute Minimum Weight Design of Framed Structures," *Quarterly Journal of Mechanics and Applied Mathematics*, No. 12, 1959.

5. Krishman, S. and K. V. Shetty: "On the Optimum Design of an I-Section Beam," *Journal of Aerospace Science*, No. 26, 1959.

6. Krishman, S. and K. V. Shetty: "A Method of Minimum Weight Design for Thin-Walled Beams," *Structural Engineer*, No. 5, 1961.

7. Haug, E.: *Minimum Weight Design of Beams with Inequality Constraints of Stress and Deflection*, Ph.D. Thesis, 1966, Kansas State University.

8. Hahn, E. W.: *Minimum Weight Elastic and Plastic Design of Beams and Simple Structures*, Ph.D. Thesis, June 1969, Purdue University.

9. Sturman, G., Albertson, L., Cornell, C. and J. Roesset: "Computer-Aided Bridge Design," *ASCE Journal of the Structural Division*, Vol. 92, No. 6, Dec. 1966.

10. Edward, C. H. and L. H. Gary: "Minimum Weight Proportions for Steel Girders," *ASCE Journal of the Structural Division*, Vol. 95, Oct. 1969.

11. Holt, E. C. and G. L. Heithecker: "Minimum Weight Properties for Steel Girders," *ASCE Journal of the Structural Division*, Vol. 95, No. 10, Oct. 1969.

12. Annamalai, N.: *Cost Optimization of Welded Plate Girders*, Ph.D. Thesis, Purdue University, 1970.

13. Lewis, A. D. M.: "Backtrack Programming in Welded Girder Design," *Proceedings of the Share-ACM-IEEE Design Automation Workshop*, July 15-18, 1968, Washington, D. C.

14. Okuba, S.: *Optimum Design of Composite Plate Girder Superstructures*, CE Thesis, 1965, M.I.T.

15. Razani, R.: *The Iterative Smoothing Method and Its Application to Minimum Cost Design of Highway Bridge Girders*, Ph.D. Thesis, 1965, Case Institute of Technology.

16. Razani, R. and G. Goble: "Optimum Design of Constant Depth Plate Girders," *ASCE Journal of the Structural Division*, Vol. 92, April 1966.

17. Goble, G. and P. V. DeSantis: "Optimum Design of Mixed Steel Composite Girders," *ASCE Journal of the Structural Division*, Vol. 92, Dec. 1966.

18. Goble, G. and P. V. DeSantis: *Girder Automated Design-I, Vol. 1 - User's Manual*, Oct. 1968.

19. Goble, G. and P. V. DeSantis: *Girder Automated Design-I, Vol. 2 - Maintenance Manual*, Oct. 1968.

20. American Association of State Highway Officials, *Standard Specifications for Highway Bridges*, Tenth Edition, 1969.

21. Gaylord, E. H. and C. N. Gaylord: *Structural Engineering Handbook*, McGraw-Hill Book Company, New York, 1968.

22. Norris, C. H. and J. B. Wilbur: *Elementary Structural Analysis*, Second Edition, McGraw-Hill Book Company, New York, 1960.

23. Shedd, T. C. and J. K. Vawter: *Theory of Simple Structures*, J. Wiley and Sons, Inc., New York, 1941.

24. Willems, N. and W. M. Lucas: *Matrix Analysis for Structural Engineers*, Prentice-Hall, Inc., New Jersey, 1968.

25. Pippard, A. J. S. and J. Baker: *The Analysis of Engineering Structures*, Fourth Edition, American Elsevier Publishing Co., New York, 1968.

26. Byars, E. F. and R. D. Snyder: Engineering Mechanics of Deformable Bodies, International Textbook Co., Pennsylvania, 1963.

27. Viest, I. M., Fountain, R. S. and R. C. Singleton: Composite Construction in Steel and Concrete, McGraw-Hill Book Co., New York, 1958.

28. McCormac, J. C.: Structural Steel Design, International Textbook Co., Pennsylvania, 1965.

29. Bresler, B., Lin, T. Y. and J. B. Scalzi: Design of Steel Structures, Second Edition, John Wiley and Sons, New York, 1968.

30. American Institute of Steel Construction: Manual of Steel Construction, Seventh Edition, 1970.

31. United States Steel Company: Highway Structures Design Handbook, Volumes 1 and 2, 1965.

32. Blodgett, O. W.: Design of Welded Structures, The James F. Lincoln Arc Welding Foundation, Ohio, 1966.

33. Slutter, R. G. and J. W. Fisher: "Fatigue Strength of Shear Connectors," Highway Research Record, No. 147, Highway Research Board, Washington, D. C., 1966.

34. Beveridge, G. S. G. and R. S. Schechter: Optimization: Theory and Practice, McGraw-Hill Book Company, New York, 1970.

35. Wilde, D. J. and C. S. Beighlter: Foundations of Optimization, Prentice-Hall, Inc., New York, 1967.

36. Pierre, D. A.: Optimization Theory with Applications, J. Wiley and Sons, Inc., New York, 1969.

37. Denn, M. M.: Optimization by Variational Methods, McGraw-Hill Book Company, New York, 1969.

38. Bellman, R.: Dynamic Programming, Princeton University Press, New Jersey, 1957.

APPENDICES

APPENDIX A

APPENDIX A

SUBROUTINE DESCRIPTIONS

The subroutines used in the program are listed and described in the following list. The listing is presented in alphabetical order.

ALLOW - returns the allowable bending stress for any location along the top and the bottom of the girder, disregarding all fatigue conditions.

ANAL - determines the design conditions (reactions, moments and shears) for a particular problem.

BASFAT - returns the allowable base metal fatigue stress at a given point on the girder.

CONDES - designs the shear connectors for a given girder.

CPDES - designs the cover plates for a given rolled section.

CPWDES - designs the required end and seal welds for each cover plate required on the girder.

DEFLEC - determines the required deflections for the final girder design.

EQSET - calculates the simple beam deflections, at the interior support points, which are used in calculating the influence lines for the reactions.

FLOPT - uses the optimization method of dynamic programming to determine the final cover plate arrangement along a girder.

ICALC - determines the moments of inertia and the distances to the lower extreme fibers for any rolled section and cover plate combination. It is also used to find the similar properties for the composite sections by treating the slab as a top cover plate.

ILINT - returns the interpolated value of any point along an influence line using straight line interpolation.

ILPROP - determines the various required properties of a given influence line including the crossover points, the positive and negative areas and the locations of the maximum and minimum ordinates.

IMPACT - determines the impact factor for any particular analysis function.

KEEP - decides whether or not to retain the given design as the present optimum.

LOAD - places all the required loads on the influence line in such a manner as to return the maximum loading conditions required.

MAIN - calls on the remaining subroutines in the correct order to produce the optimum design.

OBJECT - contains the function being minimized during the optimization of the design problem.

PRCON - prints the connector design for a given girder.

PRCOV - prints the cover plate information for a girder.

PRDEF - prints the deflections for the final girder design.

PRSEC - prints the section table for a given girder.

PRSTIF - prints the bearing stiffener information for a given girder design.

PRSTR - prints the stress table for a given girder design.

REAC - determines the reaction influence lines for a continuous girder using the moment-area theorems.

READIN - reads in all the pertinent data required for a particular design problem.

SEMIC - calculates the properties for each sub-element along the girder.

SORT - places the required cover plate thicknesses in ascending order so they can be used in the FLOPT subroutine.

STIFF - completely designs the bearing stiffeners for a given girder if they are required.

STRCHK - determines the stresses on a given girder and checks them against the allowable stresses.

STRMAX - determines the maximum design stress conditions.

TERMIN - determines the locations of the ends of the required cover plates based on the fatigue stress conditions.

THDES - calculates the required cover plate thicknesses for each subelement along the girder.

TOPT - determines the required minimum weld thickness for a fillet weld.

UNIFST - designs the girder which has a uniform steel section composed of a rolled, wide-flange section without cover plates.

APPENDIX B

APPENDIX B

SELECTED PROGRAM NOMENCLATURE

The following alphabetic list contains some of the more pertinent variables used in the program. The list basically contains all the variables which are interchanged between the program subroutines. The numbers in the parentheses immediately following the variable name give the maximum dimensions of that array within the program.

ANEG - the negative area under a given influence line.

APOS - the positive area under a given influence line.

BEAR (5,3) - the bearing stiffener array.

(I,1) - width of stiffener at support I.

(I,2) - thickness of stiffener at support I.

(I,3) - size of connecting weld for stiffener at support I.

CC (5,5) - the coefficient array for the reaction influence line calculations.

CH (80) - the distance from the neutral axis to the lower extreme fiber within a subelement for the high modulus composite section.

CL (80) - the distance from the neutral axis to the lower extreme fiber within a subelement for the low modulus composite section.

CMIN - the cost of the present optimum design.

CONSP(16,3,4) - the shear connector design array.

(J,1,I) - spacing of the J group of connectors in span I.

(J,2,I) - the starting coordinate of the J group of connectors in span I.

(J,3,I) - the final coordinate of the J group of connectors in span I.

COOR (81) - the coordinates of the analysis points.

COST (9) - the unit cost array.

(1) - cost of the rolled sections.

(2) - cost of the cover plates.

(3) - cost of the bearing stiffeners.

(4) - cost of the shear connectors.

(5) - cost of the connector installation.

(6) - fixed cost of a weld.

(7) - cost of the weld material.

(8) - cost of the weld length.

COVPL(12,6,2) - the cover plate design array.

(J,1,I) - the starting coordinate of Jth plate.

(J,2,I) - the final coordinate of Jth plate.

(J,3,I) - the thickness of Jth plate.

(J,4,I) - the width of the Jth plate.

(J,5,I) - the size of the end welds for the Jth plate.

(J,6,I) - the size of the seal weld for the Jth plate.

Note: For the COVPL array, I equals 1 for the cover plates on the top of the girder and I equals 2 for the cover plates on the bottom of the girder.

CS (80) - the distance from the neutral axis to the lower extreme fiber within a subelement for the steel section alone.

DAF (18) - the depth divided by the flange area for the rolled sections used in the design program.

DEPTH(18) - the depth of the rolled sections used in the design program.

DESDEF(20,4) - the design girder deflections.

(I,1) - the deflection at IDEF(I) due to the dead load.

(I,2) - the deflection at IDEF(I) due to the positive live load.

(I,3) - the deflection at IDEF(I) due to the superimposed dead load.

(I,4) - the deflection at IDEF(I) due to the negative live load.

DESMOM(81,4) - the design moment array.

(I,1) - the moment at I due to the dead load.

(I,2) - the moment at I due to the positive live load.

(I,3) - the moment at I due to the superimposed dead load.

(I,4) - the moment at I due to the negative live load.

DESREA(5,4) - the design reaction array.

(I,1) - the reaction at support I due to the dead load.

(I,2) - the reaction at support I due to the positive live load.

(I,3) - the reaction at support I due to the superimposed dead load.

(I,4) - the reaction at support I due to the negative live load.

DESSH (162,4) - the design shear array.

(I,1) - the shear at point I due to the dead load.

(I,2) - the shear at point I due to the positive live load.

(I,3) - the shear at point I due to the superimposed dead load.

(I,4) - the shear at point I due to the negative live load.

Note: For analysis point I, J equals 2I - 1 just to the right of I and J equals 2I - 2 just to the left of I.

DIA - the diameter of the shear connector used in the program.

DING - the impact length for a given influence line.

DIST - the distance a cover plate is cut off based on fatigue stress considerations.

EM - the modulus of elasticity which is set at 29,000 ksi in the program.

EPS (4) - the number of elements per span.

FC - the allowable concrete compressive stress.

FLTH (18) - the thickness of the flange for the rolled sections used in the design program.

FLWID (18) - the width of the flange for the rolled sections used in the design program.

FPC - the 28-day strength of the concrete.

FU - the minimum tensile strength of the steel.

FV - the allowable weld metal stress.

GDL - the weight of the steel girder.

HAUNCH - the value of the concrete haunch for the composite design.

HT - the calculated impact factor.

IDEF (30) - the analysis points at which the deflections are calculated.

IHCON (80) - the moment of inertia of a sub-element for the high modulus composite section.

ILCON (80) - the moment of inertia of a sub-element for the low modulus composite section.

IOSP (5) - the index of the support points.

IPT1 - the subelement input option.

IPT2 - the loading type input option.

IPT3 - the initial cross section input option.

IPT4 - the units input option.

IPT5 - the loading components or designation input option.

IPT6 - the optimization input option.

IPT7 - the design cycle input option.

IPT8 - the material input option.

IPT9 - the deflection input option.

IPT10 - the output option.

ISAVE (80) - the rolled section number for each subelement of the present optimum design girder.

IST - the identification number of the rolled section being designed.

ISTEEL(80) - the moment of inertia of a subelement for the steel section.

IX (18) - the moment of inertia about the x-x axis for the rolled sections used in the design program.

IY (18) - the moment of inertia about the y-y axis for the rolled sections used in the program.

JPT3 - the composite section variable.

Note: If the design is composite then JPT3 is less than 9 and if the design is noncomposite then JPT3 is greater than 9.

KOP (10) - the location of the crossover points for a given influence line.

LDIS - the live load designation.

LENGTH - the total length of the girder.

LPCM	(10)	- the boundary points of the negative moment regions along the girder.
MAXOL		- the point which is the maximum ordinate of a given influence line.
MINOL		- the point which is the minimum ordinate of a given influence line.
MN		- the number of spans along the design problem girder.
MR		- the modular ratio.
NA		- the total number of analysis points along the girder.
NAME	(18)	- the AISC name of the rolled sections used in the design program.
NCS		- the number of shear connectors per row.
NCY		- the design cycle counter.
NCYC		- the maximum number of design cycles required.
NDEF		- the total number of points at which the deflections are required.
NE		- the total number of subelements.
NOS	(80)	- the NSECT number of the rolled section in each subelement.
NS		- the total number of supports.
NSECT	(18)	- the identification number of the rolled sections used in the design program.

OI - the impact option.

OIL - the interstate loading option.

PERCEN - the allowable percentage change in the critical design moment.

PLATH (80,2) - the plate thicknesses for each subelement.

(I,1) - the thickness of the top cover plate in subelement I.

(I,2) - the thickness of the bottom plate in subelement I.

PLL - the live load factor.

PLTSAV(80,2) - the PLATH array for the current optimum design.

PLWID (80,2) - the plate width for each subelement.

(I,1) - the width of the top cover plate in subelement I.

(I,2) - the width of the bottom cover plate in subelement I.

PLWSAV(80,2) - the PLWID array for the current optimum design.

PROP (3) - the steel properties used in the design.

(1) - the allowable bending stress.

(2) - the allowable shear stress.

(3) - the steel yield point stress.

RIL (5,81,3)- the reaction influence line array.

(I,J,1) - the reaction at support I, with the unit load at point J, for the steel section.

(I,J,2) - the reaction at support I, with the unit load at point J, for the low modulus composite section.

(I,J,3) - the reaction at support I, with the unit load at point J, for the high modulus composite section.

RATIO - the ratio of the minimum stress to the maximum stress used in the fatigue calculations.

SAREA (18) - the area of the rolled sections used in the design program.

SAVE (12,6,2)- the cover plate array (COVPL) for the current optimum design.

SDL - the superimposed dead load.

SLABA - the area of the concrete slab which acts compositely with the steel section.

SLABTH - the thickness of the concrete slab.

SLABWD - the effective width of the concrete slab.

SLABWT - the weight of the concrete deck which is carried by the girder.

SLL - the sidewalk live load.

SMAX - the maximum stress at a point used to determine RATIO.

SMIN - the minimum stress at a point used to determine RATIO.

STORE (10) - the loadings for a given influence line.

STRESS(81,4,2) - the stress array.

(I,1,1) - the bottom steel stress at point I.

(I,2,1) - the top steel stress at point I.

(I,3,1) - the concrete stress at point I.

(I,4,1) - the shear stress at point I.

(I,1,2) - the bottom stress indicator.

(I,2,2) - the top stress indicator.

(I,3,2) - the concrete stress indicator.

(I,4,2) - the shear stress indicator.

Note: If the indicator is 1.0, the stress is greater than the allowable and if the indicator is 0.0, the stress is less than the allowable.

SUBLEN(80) - the subelement length array.

TCOST - the total cost of a girder design.

TDS - the total design stress at a point.

TITLE (36) - the problem title array.

TRLO (10) - the truck load components.

(1) - the first axle load.

(2) - the second axle load.

(3) - the third axle load.

(4) - the spacing between the first and second axles.

(5) - the minimum spacing between the second and the third axles.

(6) - the maximum spacing between the second and third axles.

(7) - the superimposed dead load.

(8) - the weight of the steel girder.

(9) - the impact option.

(10) - the weight of the slab carried by the steel girder.

TSWID1 - the transformed slab width for the low modulus composite section.

TSWID2 - the transformed slab width for the high modulus composite section.

UNLO (9) - the uniform load components.

(1) - the equivalent uniform live load.

(2) - the concentrated load for moment.

(3) - the concentrated load for shear.

(4) - the superimposed dead load.

(5) - the weight of the steel girder.

(6) - the interstate concentrated load.

(7) - the sidewalk live load.

(8) - the impact option.

(9) - the weight of the slab carried by the steel girder.

WEBTH (18)	- the thickness of the webs for the rolled sections used in the design program.
WMIN	- the weight of the present optimum design.
WTOT	- the total weight of a given girder design.

APPENDIX C

APPENDIX C

PROGRAM DATA

READIN Subroutine

The READIN subroutine is used to provide the program with all of the necessary data to solve a given problem. The routine does not, however, read the data cards which pertain to the section properties of the wide-flange rolled sections. These cards are read by the main program so that, if multiple problems are being solved with one run of the program, the section properties are read only once. The order of operations used in the READIN subroutine are described below. After the description, the method of data input is explained and a few samples of input data are presented.

Step 1

All of the arrays involved in the subroutine are zeroed and prepared for the given problem.

Step 2

The title cards are read and placed in the TITLE array. Two cards must be present for the title description. If a title is not desired or if the title is only on one card, then blank cards must appear in this position.

Step 3

The input parameter card, containing the input parameters IPT1 to IPT10, is read. These parameters are defined in Appendix B.

Step 4

The subroutine prints out some specified heading information if the output parameter, IPT10, dictates this operation.

Step 5

The lengths of the individual spans are read and used to determine the total length of the girder and the number of supports.

Step 6

Depending on the value of the subelement input option, IPT1, the subelement information is either read from cards or determined by the program. The quantities included in the subelement information are the number of elements per span, the analysis point coordinates, the support point indices, and the lengths of the subelements. If the design is continuous and composite, the limits of the negative moment regions are tentatively defined as the two elements on either side of the interior supports. These limits are then stored in the LPCM array.

Step 7

The data determined in steps 5 and 6 are printed out and converted to inches if required. This conversion is necessary because the program operates in the units of kips and inches.

Step 8

The steel table information for the wide-flange rolled sections used in the program is printed if the output parameter, IPT10, indicates that this information is desired.

Step 9

If the design is composite, then the composite section data are read and printed. The composite action parameter, JPT3, is calculated. If the design is noncomposite, step 9 is omitted.

Step 10

Depending on the value of IPT3, the initial cross section for each subelement along the girder is either read from cards or determined by the subroutine. With the sections specified for each subelement, SEMIC is called to evaluate the properties for each element. These properties are then printed in tabular form if required by the output parameter.

Step 11

The required loading conditions desired in the problem are read from cards. The form of the input depends on the

value of the loading parameter, IPT2, and the value of the loading components parameter, IPT5. The TRLO and the UNLO arrays are converted to the units of inches, if necessary, and are then printed in the output.

Step 12

The properties of the steel used in the problem are either read or determined by the routine. If the steel used has the ASTM designation of A36, then the PROP array is completely prepared by the subroutine. For all other steels, the yield point stress is read from a data card and the properties are calculated. The value of the modulus of elasticity for the steel is set at 29,000 ksi.

Step 13

The allowable compression stress for the concrete is computed using the 28-day concrete strength read in with the composite section data. All of the steel and concrete allowable stress properties are then listed in the output.

Step 14

If the optimization is based on cost, the unit cost data card is read. This card is not read if the optimization is based on weight. The COST array is then added to the output.

Step 15

The method of terminating the design program is either read or determined by the routine. The method which is

employed depends on the value of the design cycle parameter, IPT7. This termination method is then described in the output.

Step 16

The points at which the deflections are required are either read or determined by the subroutine. The form of this input depends on the value of the deflection parameter, IPT9. The points at which the deflections will be calculated are then listed in the output.

This completes the READIN subroutine. Once the data have been completely determined, the control of the program is returned to the main program and the design and optimization of the desired problem is started.

Method of Data Input

The method of data input for this program resembles the method used in the plate girder design program, GAD-I. It should be noted, however, that although the methods are similar, the data cards cannot be interchanged. The reason is that some of the data have a different meaning and purpose in this program. The designer should note these differences and take the appropriate precautions when working with both programs. The following is a description of the method of data input.

I. Card Set 1 - Title Cards
    A. The card set consists of two cards with the first 72 columns used on each card.
        Format (18A4/18A4)
II. Card Set 2 - Input Parameter Card
    A. The card set consists of one card with the following integer information:
        1. IPT1 - Subelement Information
            0 - 10 elements per span.
            1 - Read in the coordinates.
            2 - Read in the number of elements per span.
        2. IPT2 - Type of Loading Required
            0 - Lane, Interstate, Sidewalk and Truck Loading.

1 - Lane, Interstate and Sidewalk Loading.

2 - Truck Loading.

3. IPT3 - Initial Girder Section

0 - W33×118 used with composite design.

1 - Read the uniform section with composite design.

2 - Read the section for each subelement with composite design.

10 - W33×118 used with noncomposite design.

11 - Read the uniform section with noncomposite design.

12 - Read the section for each subelement with noncomposite design.

4. IPT4 - Units of Data

0 - Units of inches.

1 - Units of feet.

5. IPT5 - Loading Components/Designation

0 - Read in the load factor and the truck designation.

1 - Read in all the load components.

6. IPT6 - Type of Optimization

0 - Optimize based on cost.

1 - Optimize based on weight.

7. IPT7 - Design Cycles
    - 0 - Use only one cycle of design.
    - 1 - Base the design on the percentage change in the critical moment.
    - 2 - Read in the maximum number of design cycles.
8. IPT8 - Steel Used in the Design
    - 0 - Use A36 steel.
    - 1 - Read the required yield point stress.
9. IPT9 - Deflections Required
    - 0 - Determine the deflections at the span centerlines.
    - 1 - Determine the deflections at the 0.2, 0.5, and 0.8 points in each span.
    - 2 - Read in the analysis points at which the deflections are required.
10. IPT10 - Output Parameter
    - 0 - Data and the final solution.
    - 1 - Level 0 plus each cycle solution.
    - 2 - Level 1 plus each girder design.
    - 3 - Level 2 plus additional output.
    - 4 - Debugging output.

Format (10I5)

III. Card Set 3 - Span Length Card

A. The card set consists of one card which contains the lengths of the spans.

Format (4F10.4)

IV. Card Set 4 - Subelement Information

A. If IPT1 = 0, card set 4 is not required.

B. If IPT1 = 1, the card set consists of the following cards:

1. One card containing the total number of analysis points.

Format (I10)

2. The necessary number of cards, each containing seven coordinates, to completely define the total number of analysis points.

Format (7F10.2)

C. If IPT1 = 2, read the number of elements for each span.

Format (4I10)

V. Card Set 5 - Composite Section Requirements

A. If IPT3 = 10, 11, or 12, card set 5 is not required.

B. If IPT3 = 0, 1, or 3, read the following composite section requirements from one card:

1. Effective width of the slab, in.

2. Thickness of slab, in.
3. Modular Ratio as a floating point number.
4. 28-day concrete strength, psi.
5. Slab steel area, sq. in.
6. Value of the haunch, in.

Format (5F10.2,F10.4)

VI. Card Set 6 - Initial Cross Sections

A. If IPT3 = 0 or 10, card set 6 is not required.

B. If IPT3 = 1 or 11, the card set consists of one card containing the following values:

1. ID number of the rolled section.
2. Width of top cover plate, in.
3. Thickness of top cover plate, in.
4. Width of the bottom cover plate, in.
5. Thickness of the bottom cover plate, in.

Format (I10,4F10.4)

C. If IPT3 = 2 or 12, the card sets consist of one card for each subelement. The following values are contained on each card:

1. ID number of the rolled section.
2. Width of top cover plate, in.
3. Thickness of top cover plate, in.
4. Width of the bottom plate, in.

5. Thickness of the bottom plate, in.

Format (I10,4F10.4)

VII. Card Set 7 - Load Information

A. If IPT5 = 0, the card set consists of one card containing the following values:

1. Load designation
    a. Use one of the five (5) standard AASHO designations, i.e., H10, H15, H20, HS15 or HS20.
    b. This input must be right justified in the first 4 columns.
2. Load factor in decimal form.
3. Weight of the slab carried by the girder.
4. The superimposed dead load acting on the low modulus concrete section.
5. Dead weight of the girder.
6. Sidewalk live load.
7. Impact option.
    a. Use 0.0 if impact is included.
    b. Use 1.0 if impact is to be excluded from the design.
8. Interstate loading option.
    a. Use 0.0 if loading is to be considered.

b. Use 1.0 if loading is excluded.

Format (A4,F6.3,6F10.4)

B. If IPT5 = 1 and IPT2 = 0, the card set consists of two cards. The first card contains the truck loading components (TRLO) and the second card contains the uniform load components (UNLO).

1. Truck load components.
    a. First axle load in kips.
    b. Second axle load in kips.
    c. Third axle load in kips.
    d. Spacing of the first two axles.
    e. Minimum spacing between the rear axles.
    f. Maximum spacing between the rear axles.
    g. Superimposed dead load.
    h. Girder dead weight.
    i. Impact option.
    j. Weight of slab.

    Format (10F7.3)

2. Uniform load components.
    a. Uniform lane load.
    b. Concentrated load for moment.
    c. Concentrated load for shear.
    d. Superimposed dead load.

e. Girder dead weight.

f. Interstate concentrated load.

g. Sidewalk live load.

h. Impact option.

i. Weight of slab.

Format (9F8.4)

C. If IPT5 = 1 and IPT2 = 1, the card set consists of one card containing the lane loading components shown above.

D. If IPT5 = 1 and IPT2 = 2, the card set consists of one card containing the truck loading components shown above.

VIII. Card Set 8 - Material Properties

A. If IPT8 = 0, card set 8 is not required.

B. If IPT8 = 1, the card set consists of one card containing the yield point stress of the steel used in the problem. Up to 4 percent overstress can be programmed by increasing the yield point stress by the desired overstress percentage.

IX. Card Set 9 - Optimization Information

A. If IPT6 = 0, the card set consists of one card containing the following values:

1. Rolled section cost, $/lb.
2. Cover plate cost, $/lb.
3. Bearing stiffener cost, $/lb.

4. Shear connector cost, \$/lb.
5. Connector installation cost, \$/ea.
6. Fixed weld cost, \$/weld.
7. Weld material cost, \$/cu. in.
8. Weld length cost, \$/in.

   Format (9F8.2)

B. If IPT6 = 1, card set 9 is not required.

X. Card Set 10 - Design Cycle Information

A. If IPT7 = 0, card set 10 is not required.

B. If IPT7 = 1, the card set consists of one card containing the following values:

1. Maximum number of cycles.
2. Allowable percentage change in the critical moment.

   Format (I10,F10.2)

C. If IPT7 = 2, the card set consists of one card containing the number of design cycles desired.

   Format (I10)

XI. Card Set 11 - Deflection Information

A. If IPT9 = 0, card set 11 is not required.

B.. If IPT9 = 1, card set 11 is not required.

C. If IPT9 = 2, the card set consists of one card containing the following:

1. Number of points at which the deflections are required. (Max. = 20)

2. Analysis point numbers of the deflection points.
   Format (I10,20I3)

Note: When preparing the data, all integer values should be right justified and all floating point numbers should be typed according to their respective formats. All floating point fields are large enough to include a punched decimal point and this is recommended to avoid confusion.

Input Data Samples

Problem No. 1

Design a girder with three spans, each having a length of 70 ft. Design the girder for HS20 loading with a load factor of one axle per girder. Also design for impact loading and interstate loading. Use A36 steel in the design, and base the design on ten elements per span.

Design the girder compositely with the deck slab. Use concrete with a 28-day strength of 3,000 psi and a modular ratio of 8 for the bridge deck. The composite section properties include an effective slab width of 84 in., a slab thickness of 7 in., a slab steel area of 1.94 sq. in., and a haunch depth of 1.5 in.

The optimization is based on cost. The unit cost per pound of steel for all steel elements is 10 cents. The fixed weld cost is $30.00. The weld material costs $3.00 per cubic inch of material and there is no charge per length of weld. The installation cost for the shear connectors is 50 cents.

Design the girder until the change in the critical moment is less than one-half percent. Determine the deflections at the centerline of each span and use the debugging output option.

The input data for this problem are shown in Table C-1.

Table C-1: Input Data for Sample Data Problem No. 1

Card Set	Card No.	DATA CARDS
1	1	THREE SPAN TEST PROBLEM
	2	BRIDGE NO. 1 MARCH 23, 1971
2	3	0 0 0 1 0 0 1 0 0 4
3	4	70.00 70.00 70.00
5	5	84.00 7.00 8.00 3000.00 1.94 1.5
7	6	HS20 1.0 .73 .20 .15 0.0 0.0 0.0
9	7	.10 .10 .10 .10 .50 30.00 3.00 0.0
10	8	5 .5

Problem No. 2

Design a noncomposite girder for a 70 ft simple-span bridge. Design the girder in A36 steel and for an H20 AASHO loading. Use a load factor of 75 percent and do not include impact in the design. Read in the coordinates of fifteen analysis points.

Optimize the girder design based on weight and use a total of three design cycles. Determine the deflections at all points and print the solution.

The input data for this problem are shown in Table C-2.

Table C-2: Input Data for Sample Data Problem No. 2

Card Set	Card No.	DATA CARDS
1	1	SIMPLE SPAN TEST PROBLEM
	2	BRIDGE NO. 4 MARCH 23, 1971
2	3	1 0 10 1 0 1 2 0 2 0
3	4	70.00
4	5	15
	6	0.0 5.0 10.0 15.0 20.0 25.0 30.0
	7	35.0 40.0 45.0 50.0 55.0 60.0 65.0
	8	70.0
7	9	H20 .75 .73 .20 .15 0.0 1.0 0.0
10	10	3
11	11	15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

APPENDIX D

APPENDIX D

COMPUTER PROGRAM LISTING

```
                    PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
              C        MAIN DESIGN PROGRAM
000003              COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18),
                   1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),
                   2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,
                   3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),FM,SPAN(4),IOSP(5),
                   4TMOEI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLD(10),PROP(3),
                   5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),BEAR(5,3),
                   6COST(9),CONSP(16,3,4),COVPL(12,6,2)
000003              COMMON/ONE/IPT3,IPT10
000003              COMMON/TWO/LPCM
000003              COMMON/THREE/AS,NCS,IPT6
000003              COMMON/FOUR/WIDT,ICOST
000003              COMMON/SIX/FU
000003              COMMON/SEVEN/WMIN,CMIN
000003              COMMON/EIGHT/IST
000003              COMMON/NINE/ISWIT
000003              COMMON/TEN/HAUNCH
000003              COMMON/D/IDEF(20),DESDEF(20,4),NDEF,PERCEN,NCYC
000003              COMMON/A/ISAVE(80),SAVE(12,6,2),PLTSAV(80,2),PLWSAV(80,2)
000003              COMMON/B/SMEFC,ITT
000003              COMMON/E/IPT2
000003              COMMON/F/IPT7
000003              REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH
000003              DIMENSION TITLE(36), EPS(4), TIL(81,3), STORE(16), LPCM(10)
000003              DIMENSION IMST(4)
000003              INTEGER TYPE
000003              INTEGER EPS,TITLE
              C        READ IN THE BEAM TABLE DATA
000003              DO 1000 I = 1, 18
000005              NSECT(I) = I
000006         1000 READ (5,1001) NAME(I),SAREA(I),DEPTH(I),FLWID(I),FLTH(I),WEBTH(I),
                   1DAF(I),IX(I),IY(I)
000036         1001 FORMAT (3X,A7,6F10.4/2F10.2)
              C        CALL ON READIN SUBROUTINE TO READ AND CHECK DATA AFTER ZEROING
              C        OUT THE REQUIRED MATRICES FOR A NEW PROBLEM
000036          900 CALL READIN
000037              SMEFC = 0.0
              C        DETERMINE FU FOR FATIGUE CALCULATIONS
000040              IF (ABS(PROP(3)-36.) .LT. 1.45 ) FU = 58.
000046              IF (ABS(PROP(3)-50.) .LT. 2.41 ) FU = 65.
000054              IF (ABS(PROP(3)-42.) .LT. 1.70 ) FU = 60.
000062              IF (ABS(PROP(3)-45.) .LT. 1.85 ) FU = 60.
000070              IF (ABS(PROP(3)-60.) .LT. 2.50 ) FU = 75.
000076              IF (ABS(PROP(3)-55.) .LT. 2.21 ) FU = 70.
              C        CALL ON ANAL TO ANALYSIZE THE GIRDER
000104              MN = NS - 1
000106              NCY = 1
000107              CALL ANAL
              C        BEGIN THE DESIGN CYCLE
000110           80 WMIN = 99999999.
000112              CMIN = 99999999.
000113              IF (IPT10 .GE. 2) WRITE (6,81) NCY
000123           81 FORMAT (1H1,20X,10HBEGIN THE ,I2,16H CYCLE OF DESIGN//)
              C        DETERMINE THE UNIFORM SECTION DESIGN
000123              CALL UNIFST
```

```
000124          IB = IST + 1                                                     MAIN  57
000126          IF (IB .EQ. 19) GO TO 1080                                       MAIN  58
         C         BEGIN THE DESIGN LOOP FOR THE LIGHTER SECTIONS                MAIN  59
000130          DO 1020 JJ = IB, 18                                              MAIN  60
000131          IST = JJ                                                         MAIN  61
000132          DO 1030 K = 1, NF                                                MAIN  62
         C         DETERMINE THE SUBELEMENT PLATE THICKNESSES                    MAIN  64
000134     1030 NOS(K) = IST                                                     MAIN  63
000140          CALL THDES                                                       MAIN  65
000141          IF (ISWIT .EQ. 1) GO TO 1020                                     MAIN  66
000143          CALL SEMIC                                                       MAIN  67
000144          IF (IPT10 .GT. 2) CALL PRSEC(NE,NOS,NAME,PLWID,PLATH,ISTEEL,CS,  MAIN  68
               1ILCON,CL,IHCON,CH,COOR,SUBLEN,0,NCY)                             MAIN  69
         C         DESIGN THE COVER PLATES                                       MAIN  70
000165          CALL CPDES                                                       MAIN  71
000166          CALL SEMIC                                                       MAIN  72
         C         DESIGN THE WELDS                                              MAIN  73
000167          CALL CPWDES                                                      MAIN  74
         C         DESIGN THE STIFFENERS                                         MAIN  75
000170          CALL STIFF(IDSP,NS,DESREA,NOS,FLTH,FLWID,WEBTH,DEPTH,PROP,EM,    MAIN  76
               1BEAR)                                                            MAIN  77
         C         DESIGN THE SHEAR CONNECTOR SPACING                            MAIN  78
000203          CALL CONDES                                                      MAIN  79
         C         DETERMINE THE WEIGHT AND COST                                 MAIN  80
000204          CALL OBJECT                                                      MAIN  81
         C         STORE THE DESIGN IF IT IS THE PRESENT OPTIMUM                 MAIN  82
000205          CALL KEEP(NE,NOS,PLATH,PLWID,COVPL,CMIN,WMIN,IPT6)               MAIN  83
000215          IF (IPT10 .LE. 1) GO TO 1020                                     MAIN  84
000220          CALL PRSEC(NE,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CL,IHCON,CH,  MAIN  85
               1COOR,SUBLEN,0,NCY)                                               MAIN  86
000236          CALL PRCOV (COVPL)                                               MAIN  87
000240          WRITE (6,700)                                                    MAIN  88
000244      700 FORMAT (1H1)                                                     MAIN  89
000244          CALL PRSTIF(NS,BEAR)                                             MAIN  90
000246          CALL PRCON(NS,CONSP,NCS)                                         MAIN  91
000251          WRITE (6,1518) WTOT, TCOST                                       MAIN  92
000261     1518 FORMAT (1H0,10X,15HTOTAL WEIGHT = ,F15.6,5X,13HTOTAL COST = ,    MAIN  93
               1F15.2)                                                           MAIN  94
000261          IF (IPT6 .GT. .5) GO TO 6090                                     MAIN  95
000265          WRITE (6,1582) CMIN                                              MAIN  96
000272     1582 FORMAT (1H0,30X,7HCMIN = ,F12.2)                                 MAIN  97
000272          GO TO 1020                                                       MAIN  98
000273     6090 WRITE (6,3582) WMIN                                              MAIN  99
000301     3582 FORMAT (1H0,30X,7HWMIN = ,F12.2)                                 MAIN 100
000301     1020 CONTINUE                                                         MAIN 101
000303     1080 CONTINUE                                                         MAIN 102
         C         PLACE THE OPTIMUM DESIGN IN THE WORKING ARRAYS                MAIN 103
000303          DO 1 I = 1, NA                                                   MAIN 104
000305        1 NOS(I) = ISAVE(I)                                                MAIN 105
000311          IST = NOS(III)                                                   MAIN 106
000313          DO 2 I = 1, NF                                                   MAIN 107
000314          DO 2 J = 1, 2                                                    MAIN 108
000315          PLATH(I,J) = FLTSAV(I,J)                                         MAIN 109
000322        2 PLWID(I,J) = PLWSAV(I,J)                                         MAIN 110
000331          DO 3 I = 1, 12                                                   MAIN 111
000333          DO 3 J = 1, 6                                                    MAIN 112
000334          DO 3 K = 1, 2                                                    MAIN 113
000335        3 COVPL(I,J,K) = SAVE(I,J,K)                                       MAIN 114
```

```
000355          CALL SEMIC                                                   MAIN 115
000356          IF (MN .EQ. 1) GO TO 100                                     MAIN 116
          C        DETERMINE THE NEW STEEL DEAD WEIGHT                       MAIN 117
000360          IF (IPT2 .LT. 2) UNLO(5) = WMIN / (LENGTH * 1000.)           MAIN 118
000366          IF (IPT2 .NE. 1) TRLO(8) = WMIN / (LENGTH * 1000.)           MAIN 119
          C        REANALYZE THE STRUCTURE IF IT IS CONTINUOUS               MAIN 120
000373          CALL ANAL                                                    MAIN 121
000374          GO TO 200                                                    MAIN 122
000375      100 CONTINUE                                                     MAIN 123
          C        REANALYZE THE SIMPLE SPAN PROBLEM                         MAIN 124
000375          IF (JPT3 .LT. 9) GO TO 40                                    MAIN 125
000400          DLO = TRLO(7) + TRLO(8) + TRLO(10)                           MAIN 126
000403          IF (IPT2 .EQ. 2) GO TO 41                                    MAIN 127
000405          IF (DLO.LT.UNLO(4)+UNLO(5)+UNLO(9)) DLO=UNLO(4)+UNLO(5)+UNLO(9)   MAIN 128
000414          GO TO 41                                                     MAIN 129
000415       40 DLO = TRLO(8) + TRLO(10)                                     MAIN 130
000417          IF (IPT2 .EQ. 2) GO TO 41                                    MAIN 131
000421          IF (DLO .LT. UNLO(5)+UNLO(9)) DLO = UNLO(5) + UNLO(9)        MAIN 132
000426       41 IF (IPT2 .LT. 2) UNLO(5) = WMIN / (LENGTH * 1000.)           MAIN 133
000434          IF (IPT2 .NE. 1) TRLO(8) = WMIN / (LENGTH * 1000.)           MAIN 134
000441          IF (JPT3 .LT. 9) GO TO 42                                    MAIN 135
000444          DLN = TRLO(7) + TRLO(8) + TRLO(10)                           MAIN 136
000447          IF (IPT2 .EQ. 2) GO TO 43                                    MAIN 137
000451          IF (DLN.LT.UNLO(4)+UNLO(5)+UNLO(9)) DLN=UNLO(4)+UNLO(5)+UNLO(9)   MAIN 138
000460          GO TO 43                                                     MAIN 139
000461       42 DLN = TRLO(8) + TRLO(10)                                     MAIN 140
000463          IF (IPT2 .EQ. 2) GO TO 43                                    MAIN 141
000465          IF (DLN .LT. UNLO(5) + UNLO(9)) DLN = UNLO(5) + UNLO(9)      MAIN 142
000472       43 FI = DLN / DLO                                               MAIN 143
000474          DO 44 I = 1, NA                                              MAIN 144
000476       44 DESMOM(I,1) = DESMOM(I,1) * FI                               MAIN 145
000503          DESREA(1,1) = DESREA(1,1) * FI                               MAIN 146
000504          DESREA(2,1) = DESREA(2,1) * FI                               MAIN 147
000505          IB = 2 * NA - 2                                              MAIN 148
000507          DO 45 I = 1, IB                                              MAIN 149
000511       45 DESSH(I,1) = DESSH(I,1) * FI                                 MAIN 150
000516          IF (IPT10 .EQ. 0) GO TO 200                                  MAIN 151
000517          WRITE (6,70)                                                 MAIN 152
000523       70 FORMAT (1H1,19HDESIGN REACTIONS...//)                        MAIN 153
000523          DO 71 I = 1, NS                                              MAIN 154
000525       71 WRITE (6,72) I,DESREA(I,1),DESREA(I,2),DESREA(I,3),DESREA(I,4)   MAIN 155
000545       72 FORMAT (I5,4(F15.4,3X))                                      MAIN 156
000545          WRITE (6,73)                                                 MAIN 157
000550       73 FORMAT (1H0,17HDESIGN MOMENTS...//)                          MAIN 158
000550          DO 74 I = 1, NA                                              MAIN 159
000552       74 WRITE (6,72) I,DESMOM(I,1),DESMOM(I,2),DESMOM(I,3),DESMOM(I,4)   MAIN 160
000572          WRITE (6,75)                                                 MAIN 161
000575       75 FORMAT (1H0,16HDESIGN SHEARS...//)                           MAIN 162
000575          KJ = 2 * NA - 2                                              MAIN 163
000577          DO 76 I = 1, KJ                                              MAIN 164
          C        DETERMINE ALL OF THE STRESSES                             MAIN 166
000601       76 WRITE (6,72) I,DESSH(I,1),DESSH(I,2),DESSH(I,3),DESSH(I,4)   MAIN 165
000621      200 CALL STRCHK                                                  MAIN 167
          C        DESIGN THE STIFFENERS                                     MAIN 168
000622          CALL STIFF(IDSP,NS,DESREA,NOS,FLTH,FLWID,WEBTH,DEPTH,PROD,EM,     MAIN 169
               1BEAM)                                                        MAIN 170
          C        DESIGN THE SHEAR CONNECTOR SPACING                        MAIN 171
000635          CALL CONDES                                                  MAIN 172
```

```
        C          DETERMINE THE WEIGHT AND COST
000636           CALL OBJECT
000637           IF (IPT10 .EQ. 0)GO TO 601
000640           CALL PRSEC(NF,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CI,THCON,CH,
                1COOR,SUBLEN,1,NCY)
000657           CALL PRCOV (COVPL)
000661           CALL PRSTR (STRESS,NA)
000663           CALL PRSTIF (NS,BEAR)
000665           CALL PRCON (NS,CONSP,NCS)
000670           IF (IPT6 .GT. .5) GO TO 10
000674           WRITE (6,11) TCOST
000701        11 FORMAT (1H0,10X,25HTHE COST OF THE GIRDER = ,F12.2)
000701           GO TO 16
000702        10 WRITE (6,15) WTOT
000710        15 FORMAT (1H0,10X,27HTHE WEIGHT OF THE GIRDER = ,F12.2)
000710        16 CONTINUE
000710       601 CONTINUE
        C          DETERMINE THE NEXT DESIGN CYCLE
000710           IF (IPT7 .NE. 2) GO TO 20
000712           IF (NCY .EQ. NCYC) GO TO 25
000714           NCY = NCY + 1
000715           GO TO 80
000716        20 IF (IPT7 .EQ. 0) GO TO 25
000717           D1 = DESMOM(ITT,1) + DESMOM(ITT,3) + DESMOM(ITT,4)
000723           IF (KN .EQ. 1) D1 = DESMOM(ITT,1)
000727           IF (KN .EQ.1 .AND. JPT3.GT.9) D1 = DESMOM(ITT,1) + DESMOM(ITT,2)
000741           P1 = ABS((SMEFC - D1) / D1)
000745           IF (P1 .LT. PERCEN/100.) GO TO 25
000750           IF (NCY .EQ. NCYC) GO TO 25
000752           NCY = NCY + 1
000753           GO TO 80
000754        25 IST = NOS(ITT)
        C          DETERMINE THE SUBELEMENT PLATE THICKNESSES FOR THE FINAL DESIGN
000756           CALL THDES
000757           CALL SEMIC
        C          DESIGN THE FINAL COVER PLATES
000760           CALL CPDES
000761           CALL SEMIC
        C          DESIGN THE FINAL WELDS
000762           CALL CPWDES
        C          DESIGN THE FINAL STIFFENERS
000763           CALL STIFF(IOSP,NS,DESREA,NOS,FLTH,FLWID,WEBTH,DEPTH,PROP,FM,
                1BEAR)
        C          DESIGN THE FINAL SHEAR CONNECTOR SPACING
000776           CALL CONDES
        C          DETERMINE THE FINAL WEIGHT AND COST
000777           CALL OBJECT
        C          CALCULATE THE FINAL STRESSES
001000           CALL STRCHK
        C          CALCULATE THE REQUIRED DEFLECTIONS
001001           CALL DEFLEC
001002           CALL PRSEC(NF,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CI,THCON,CH,
                1COOR,SUBLEN,2,NCY)
001021           CALL PRCOV(COVPL)
001023           CALL PRSTR (STRESS,NA)
001025           CALL PRSTIF (NS,BEAR)
001027           CALL PRCON (NS,CONSP,NCS)
001032           IF (IPT6 .GT. .5) GO TO 28
```

```
001036          WRITE (6,11) TCOST
001043          GO TO 29
001044       28 WRITE (6,15) WTOT
001052       29 CONTINUE
001052          WRITE (6,21)
001056       21 FORMAT (1H1)
001056          CALL PROEF (NDEF,DESDEF,NS,SPAN,IDEF,IPT2,UNLO(7))
        C          RETURN TO THE READIN STEP FOR A NEW PROBLEM
001066          GO TO 900
001067          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
004006

UNUSED COMPILER SPACE
036500
```

```
                FUNCTION ALLOW(DM,FLANGE,LPCM,IOSP,IA,NS,PROP,COOR,DESMOM,SUBLEN,
               1EM,KEY)
        C          DETERMINES THE ALLOWABLE STRESS AT A SUBELEMENT
000017          DIMENSION LPCM(10),IOSP(5),PROP(3),COOR(81),SUBLEN(80),
               1DESMOM(81,4)
000017          IF (EM .LT. -.01) GO TO 617
000021          ALLOW = PROP(1)
000023          RETURN
000024      617 DO 620 I = 1, 10, 2
000026          II = I
000027          IF (IA .GE. LPCM(I) .AND. IA .LE. LPCM(I+1)) GO TO 621
000040      620 CONTINUE
000042      621 IB = LPCM(II)
000045          IC = LPCM(II+1)
000046          DO 630 I = 2, NS
000050          IF (IOSP(I) .GE. IB .AND. IOSP(I) .LE. IC) GO TO 631
000062          GO TO 630
000062      631 ID = IOSP(I)
000065          GO TO 635
000065      630 CONTINUE
000070      635 IF (IA .LT. ID) GO TO 640
000072          DUS = COOR(IC)-COOR(ID)-(DESMOM(IC,1)*SUBLEN(IC))/
               1(DESMOM(IC+1,1)-DESMOM(IC,1))
000106          GO TO 641
000107      640 DUS = COOR(ID)-COOR(IB)-(DESMOM(IB,1)*SUBLEN(IB-1))/
               1(DESMOM(IB-1,1)-DESMOM(IB,1))
000125      641 IF (DUS/FLANGE .GE. 216./SQRT(PROP(3))) GO TO 642
000141          RATIO = DUS/FLANGE
000142          GO TO 643
000142      642 RATIO = 216./SQRT(PROP(3))
000150      643 PAS = PROP(1) - .1671791711 * RATIO * RATIO * PROP(3)**2/EM
000157          IF (KEY .EQ. 1) GO TO 650
000165          ALLOW = PAS
000166          RETURN
000167      650 IF (1.2*PAS .LE. PROP(1)) GO TO 644
000173          ALLOW = 1.2 * PAS
```

```
000175           GO TO 645                                                                ALLO   37
000175       644 ALLOW = PROP(1)                                                          ALLO   38
000177       645 CONTINUE                                                                 ALLO   39
000177           RETURN                                                                   ALLO   40
000201           END                                                                      ALLO   41

SUBPROGRAM LENGTH
000235

UNUSED COMPILER SPACE
042300


                 SUBROUTINE ANAL                                                          ANAL    1
         C          CALCULATES THE DESIGN REACTIONS, MOMENTS, AND SHEARS                  ANAL    2
         C          FOR A GIVEN GIRDER                                                    ANAL    3
000002           COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18),        ANAL    4
                1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),               ANAL    5
                2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,            ANAL    6
                3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),FM,SPAN(4),IOSP(5),        ANAL    7
                4TMOMI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLO(10),PROP(3),ANAL    8
                5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4)                  ANAL    9
000002           COMMON/ONE/JPT3,IPT10                                                    ANAL   10
000002           COMMON/TWO/LPCM                                                          ANAL   11
000002           COMMON/TEN/HAUNCH                                                        ANAL   12
000002           COMMON/F/IPT2                                                            ANAL   13
000002           REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                                  ANAL   14
000002           DIMENSION TITLE(36), EPS(4), TIL(81,3), STORE(10), LPCM(10)              ANAL   15
000002           INTEGER TYPE                                                             ANAL   16
000002           INTEGER EPS,TITLE                                                        ANAL   17
         C          ZERO OUT THE DESIGN ARRAYS                                            ANAL   18
000002           DO 500 J = 1, 4                                                          ANAL   19
000004           DO 501 I = 1, 81                                                         ANAL   20
000005       501 DESMOM(I,J) = 0.0                                                        ANAL   21
000013           DO 502 I = 1, 5                                                          ANAL   22
000014       502 DESREA(I,J) = 0.0                                                        ANAL   23
000021           DO 503 I = 1, 162                                                        ANAL   24
000022       503 DESSH(I,J) = 0.0                                                         ANAL   25
000030       500 CONTINUE                                                                 ANAL   26
000032           MN = NS - 1                                                              ANAL   27
000034           IF (MN .NE. 1) GO TO 3000                                                ANAL   28
         C          DETERMINE THE REACTION INFLUENCE LINES FOR THE SIMPLE SPAN            ANAL   29
000035           DO 3001 I = 1, NA                                                        ANAL   30
000036           RIL(1,I,1) = (LENGTH - COOR(I)) / LENGTH                                 ANAL   31
000043      3001 RIL(2,I,1) = 1.0 - RIL(1,I,1)                                            ANAL   32
000052           DO 3002 I = 2, 3                                                         ANAL   33
000053           DO 3002 J = 1, NA                                                        ANAL   34
000054           RIL(1,J,I) = RIL(1,J,1)                                                  ANAL   35
000062      3002 RIL(2,J,I) = RIL(2,J,1)                                                  ANAL   36
000073           IF (IPT10 .LT. 3) GO TO 1002                                             ANAL   37
000075           GO TO 3003                                                               ANAL   38
         C          CALL ON REAC TO DETERMINE THE CONTINUOUS REACTION LINES               ANAL   39
000076      3000 CALL REAC                                                                ANAL   40
         C          PRINTOUT THE REACTION INFLUENCE LINE IF REQUIRED                      ANAL   41
```

```
000077           IF (IPT10 .LT.3) GO TO 1002                                       ANAL  42
000102           IF (JPT3 .GE. 9) GO TO 1003                                       ANAL  43
000105           WRITE (6,1010)                                                    ANAL  44
000110     1010 FORMAT (1H1,46X,23HREACTION INFLUENCE LINE//)                      ANAL  45
000110           DO 1040 K = 1, 3                                                  ANAL  46
000112           IF (K .EQ. 1) WRITE (6,1050)                                      ANAL  47
000117           IF (K .EQ. 2) WRITE (6,1051)                                      ANAL  48
000125           IF (K .EQ. 3) WRITE (6,1052)                                      ANAL  49
000133     1050 FORMAT (1H0,30X,29HREACTION LINE FOR STEEL ALONE//)                ANAL  50
000133     1051 FORMAT (1H1,30X,46HREACTION LINE FOR LOW MODULUS COMPOSITE ACTION/ANAL  51
               1/)                                                                 ANAL  52
000133     1052 FORMAT(1H1,30X,47HREACTION LINE FOR HIGH MODULUS COMPOSITE ACTION/ANAL  53
               1/)                                                                 ANAL  54
000133           DO 1041 J = 1, NA                                                 ANAL  55
000135     1041 WRITE (6,1053) (RIL(I,J,K), I = 1, NS)                             ANAL  56
000155     1053 FORMAT (31X,5(F12.8,2X))                                           ANAL  57
000155     1040 CONTINUE                                                           ANAL  58
000157           GO TO 1002                                                        ANAL  59
000157     1003 WRITE (6,1010)                                                     ANAL  60
000163           WRITE (6,1050)                                                    ANAL  61
000167           DO 2060 I = 1, NS                                                 ANAL  62
000171           RIL(I,J,2) = RIL(I,J,1)                                           ANAL  63
000176     2060 RIL(I,J,3) = RIL(I,J,1)                                            ANAL  64
000202           DO 1060 J = 1, NA                                                 ANAL  65
000203     1060 WRITE (6,1053) (RIL(I,J,1), I = 1, NS)                             ANAL  66
000221           WRITE (6,1054)                                                    ANAL  67
000224     1054 FORMAT (1H0,20X,43HTHERE IS NO COMPOSITE ACTION IN THE DESIGN.)    ANAL  68
000224           GO TO 1002                                                        ANAL  69
000225     3003 WRITE (6,3004)                                                     ANAL  70
000231     3004 FORMAT (1H1,28X,39HREACTION INFLUENCE LINE FOR SIMPLE BEAM//)      ANAL  71
000231           DO 3005 I = 1, NA                                                 ANAL  72
000233     3005 WRITE (6,1053) (RIL(J,I,1), J = 1, NS)                             ANAL  73
000251           IF (JPT3 .GT. 9) WRITE (6,1054)                                   ANAL  74
          C         PRINT OUT THE LOADING DESCRIPTION                              ANAL  75
000257     1002 IF (IPT10 .LT. 3) WRITE (6,6000)                                   ANAL  76
000265     6000 FORMAT (1H1)                                                       ANAL  77
000265           IF (IPT10 .GE. 3) WRITE (6,1080)                                  ANAL  78
000274     1080 FORMAT (1H1,10X,34HDISCRIPTION OF ANALYSIS RESULTS.../11X,38HCOLUMANAL  79
               1N  1 - DEAD LOAD ON STEEL SECTION/11X,30HCOLUMN  2 - POSITIVE LANEANAL  80
               2 LOAD/11X,30HCOLUMN  3 - NEGATIVE LANE LOAD/11X,39HCOLUMN  4 - POSANAL  81
               3ITIVE INTERSTATE LOADING/11X,39HCOLUMN  5 - NEGATIVE INTERSTATE LOANAL  82
               4ADING/11X,31HCOLUMN  6 - POSITIVE TRUCK LOAD/11X,31HCOLUMN  7 - NEANAL  83
               5GATIVE TRUCK LOAD/11X,34HCOLUMN  8 - POSITIVE SIDEWALK LOAD/11X,   ANAL  84
               634HCOLUMN  9 - NEGATIVE SIDEWALK LOAD/11X,64HCOLUMN 10 - SUPERIMPOANAL  85
               7SED DEAD LOAD ON LOW MODULUS CONCRETE SECTION//)                   ANAL  86
000274           IF (IPT10 .NE. 0) WRITE (6,6001)                                  ANAL  87
000301     6001 FORMAT (1H0,10X,30HDISCRIPTION OF DESIGN LOADS...//11X,            ANAL  88
               135HCOLUMN 1 - LOADING DUE TO DEAD LOAD/11X,41HCOLUMN 2 - LOADING FANAL  89
               2OR POSITIVE LIVE LOAD/11X,48HCOLUMN 3 - LOADING DUE TO SUPERIMPOSEANAL  90
               3D DEAD LOAD/11X,41HCOLUMN 4 - LOADING FOR NEGATIVE LIVE LOAD//)    ANAL  91
          C         LOAD THE REACTION INFLUENCE LINES                              ANAL  92
000301           IF (IPT10 .GE. 3) WRITE (6,1090)                                  ANAL  93
000310     1090 FORMAT (1H0,22HREACTION LOADINGS.....///)                          ANAL  94
000310           DO 1100 I = 1, NS                                                 ANAL  95
000312           NP = I                                                            ANAL  96
000313           DO 1101 K = 1, NA                                                 ANAL  97
000315           DO 1101 J = 1, 3                                                  ANAL  98
000316     1101 TI(K,J) = RIL(I,K,J)                                               ANAL  99
```

```
              C         CALL THE LOAD SUBROUTINE
000333                 CALL LOAD(TIL,TRLO,UNLO,SUBLEN,NA,IPT2,LENGTH,COOR,NP,4HREAC,NS,
                      1IOSP,SPAN,STORE)
              C         PRINT THE REACTION LOADINGS IF REQUIRED
000351                 IF (IPT10 .GE. 3) WRITE(6,1103) I,(STORE(JJ),JJ=1,10)
000370            1103 FORMAT (I5, 10F12.3)
              C         DETERMINE THE DESIGN REACTIONS
000370                 DESREA(I,1) = STORE(1)
000372                 DESREA(I,3) = STORE(10)
000374                 PR = 0.0
000375                 TR = 0.0
000376                 DO 1100 KK = 2, 7
000377                 IF (STORE(KK) .GE. 0.0) GO TO 8100
000401                 RR = STORE(9) + STORE(KK)
000403                 IF (ABS(RR) .LT. TR) GO TO 1100
000406                 TR = ABS(RR)
000407                 DESREA(I,4) = RR
000410                 GO TO 1100
000411            8100 RR = STORE(8) + STORE(KK)
000414                 IF (ABS(RR) .LT. PR) GO TO 1100
000417                 PR = ABS(RR)
000420                 DESREA(I,2) = RR
000421            1100 CONTINUE
              C         PRINT THE DESIGN REACTIONS IF REQUIRED
000426                 IF (IPT10 .EQ. 0) GO TO 100
000427                 WRITE (6,1115)
000432            1115 FORMAT (1H0,19HDESIGN REACTIONS...//)
000432                 DO 1118 I = 1, NS
000434            1118 WRITE (6,1120) I,DESREA(I,1),DESREA(I,2),DESREA(I,3),DESREA(I,4)
000454            1120 FORMAT (I5,4(F15.4,3X))
000454                 IF (IPT3 .GT. 9) WRITE(6,6010)
000462            6010 FORMAT (10X,57HTHE THIRD COLUMN IS ZERO BECAUSE DESIGN IS NON-COMP
                      1OSITE.)
              C         CONSTRUCT THE MOMENT INFLUENCE LINES
000462             100 IF (IPT10 .GE. 3) WRITE (6,1143)
000471            1143 FORMAT (1H0,18HMOMENT LOADINGS...//)
000471                 DO 1130 I = 1, 3
000473                 TIL(1,I) = 0.0
000476            1130 TIL(NA,I) = 0.0
000502                 DO 1140 I = 2, NF
000504                 DO 1150 K = 1, 3
000505                 DO 1135 J = 2, NF
000506                 WP = 0.0
000507                 SP = 0.0
000510                 IF (J .LT. I) SP = COOR(I) - COOR(J)
000514                 JJ = 1
000515            1141 KK = IOSP(JJ)
000517                 IF (I .GT. KK) WP = WP + RIL(JJ,J,K) * (COOR(I) - COOR(KK))
000532                 IF (KK .GT. I) GO TO 1142
000536                 JJ = JJ + 1
000537                 GO TO 1141
000540            1142 TIL(J,K) = WP - SP
000545            1135 CONTINUE
000547            1150 CONTINUE
              C         CALL ON THE LOAD SUBROUTINE
000551                 CALL LOAD (TIL,TRLO,UNLO,SUBLEN,NA,IPT2,LENGTH,COOR,I,4HMOMT,NS,
                      1IOSP,SPAN,STORE)
              C         PRINT THE MOMENT LOADINGS IF REQUIRED
```

```
000566         IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)        ANAL 158
        C          DETERMINE THE DESIGN MOMENTS                                     ANAL 159
000605         DESMOM(I,1) = STORE(1)                                               ANAL 160
000607         DESMOM(I,3) = STORE(10)                                              ANAL 161
000611         PR = 0.0                                                             ANAL 162
000612         TR = 0.0                                                             ANAL 163
000613         DO 1140 KK = 2, 7                                                    ANAL 164
000614         KB = 2 * (KK / 2 )                                                   ANAL 165
000616         IF(ABS(STORE(KK)).LT..00001.AND.KB.EQ.KK) RR=STORE(8)+STORE(KK)      ANAL 166
000632         IF(ABS(STORE(KK)).LT..00001.AND.KB.NE.KK) RR=STORE(9)+STORE(KK)      ANAL 167
000647         IF (STORE(KK) .GT. 0.0) RR = STORE(8) + STORE(KK)                    ANAL 168
000653         IF (STORE(KK) .LT. 0.0) RR = STORE(9) + STORE(KK)                    ANAL 169
000657         IF (RR .GE. 0.0) GO TO 5240                                          ANAL 170
000661         IF (ABS(RR) .LT. TR) GO TO 5140                                      ANAL 171
000664         TR = ABS(RR)                                                         ANAL 172
000665         DESMOM(I,4) = RR                                                     ANAL 173
000666         GO TO 5140                                                           ANAL 174
000667    5240 IF (ABS(RR) .LT. PR) GO TO 5140                                      ANAL 175
000673         PR = ABS(RR)                                                         ANAL 176
000674         DESMOM(I,2) = RR                                                     ANAL 177
000675    5140 CONTINUE                                                             ANAL 178
000677    1140 CONTINUE                                                             ANAL 179
        C          PRINT THE DESIGN MOMENTS IF REQUIRED                             ANAL 180
000702         IF (IPT10 .EQ. 0) GO TO 101                                          ANAL 181
000703         WRITE (6,1156)                                                       ANAL 182
000706    1156 FORMAT (1H0,17HDESIGN MOMENTS...//)                                  ANAL 183
000706         DO 1157 I = 1, NA                                                    ANAL 184
000710    1157 WRITE (6,1120) I,DESMOM(I,1),DESMOM(I,2),DESMOM(I,3),DESMOM(I,4)     ANAL 185
000730         IF (JPT3 .GT. 8) WRITE (6,6010)                                      ANAL 186
        C          DETERMINE THE NEW LPCM VECTOR AND CORRECT THE I VALUES           ANAL 187
000736     101 CONTINUE                                                             ANAL 188
000736         IF (MN .EQ. 1) GO TO 1173                                            ANAL 189
000740         J = 1                                                                ANAL 190
000741         DO 1160 I = 1, NA                                                    ANAL 191
000743         NJ = I + 1                                                           ANAL 192
000745         IF (NJ .GT. NA) GO TO 1170                                           ANAL 193
000750         SUM1 = DESMOM(I,1)                                                   ANAL 194
000751         SUM2 = DESMOM(NJ,1)                                                  ANAL 195
000752         IF (SUM1.LT.0.0 .AND. SUM2.GT.0.0) GO TO 1161                        ANAL 196
000761         IF (SUM2.LT.0.0 .AND. SUM1.GT.0.0) GO TO 1162                        ANAL 197
000770         GO TO 1160                                                           ANAL 198
000770    1161 LPCM(J) = I                                                          ANAL 199
000772         J = J + 1                                                            ANAL 200
000774         GO TO 1160                                                           ANAL 201
000774    1162 LPCM(J) = NJ                                                         ANAL 202
000776         J = J + 1                                                            ANAL 203
001000    1160 CONTINUE                                                             ANAL 204
001003    1170 IF (JPT3 .GE. 10) GO TO 1173                                         ANAL 205
001006         CALL SEMIC                                                           ANAL 206
001007    1173 CONTINUE                                                             ANAL 207
        C          CONSTRUCT AND LOAD THE SHEAR INFLUENCE LINES TO THE              ANAL 208
        C          LEFT AND RIGHT OF THE ANALYSIS POINTS                            ANAL 209
001007         IF (IPT10 .GE. 3) WRITE (6,1300)                                     ANAL 210
001016    1300 FORMAT (1H0,19HSHEAR LOADINGS.....//)                                ANAL 211
001016         MN = NS - 1                                                          ANAL 212
001020         DO 1201 I = 1, NA                                                    ANAL 213
001022         KL = 0                                                               ANAL 214
001023         IF (I .NE. 1) GO TO 1206                                             ANAL 215
```

```
        C          CONSTRUCT THE LINE FOR POINT 1
001025             KJ = 1
001026             DO 1202 K = 1, 3
001027             TIL(1,K) = -1.0
001032             DO 1203 J = 2, NA
001034        1203 TIL(J,K) = -RIL(1,I,K)
001046        1202 CONTINUE
        C          LOAD THE RIGHT POINT  WITH THE LOAD SUBROUTINE
001050             CALL LOAD(TIL,TRLO,UNLO,SUBLEN,NA,IPT2,LENGTH,COOR,1,4HSHER,NS,
                  1IDSP,SPAN,STORE)
        C          PRINT THE SHEAR LOADINGS IF REQUIRED
001066             IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)
001105             GO TO 1220
001106        1205 IF (I .NE. NA) GO TO 1210
        C          CONSTRUCT THE LINE FOR POINT NA
001110             SP = 1.0
001112             KJ = 2 * NA - 2
001114             DO 1207 K = 1, 3
001115             TIL(NA,K) = 1.0
001121             DO 1208 J = 1, NE
001122             WP = 0.0
001123             DO 1209 JJ = 1, MN
001125        1209 WP = WP + RIL(JJ,J,K)
001135        1208 TIL(J,K) = SP - WP
001144        1207 CONTINUE
        C          LOAD THE LINE WITH THE LOAD SUBROUTINE
001146             CALL LOAD(TIL,TRLO,UNLO,SUBLEN,NA,IPT2,LENGTH,COOR,NA,4HSHER,NS,
                  1IDSP,SPAN,STORE)
        C          PRINT THE SHEAR LOADINGS IF REQUIRED
001164             IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)
001203             GO TO 1220
001204        1210 KL = 1
001205             KJ = 2 * I - 2
        C          CONSTRUCT THE LINE TO THE LEFT OF THE POINT
001207             DO 1214 K = 1, 3
001211             DO 1215 J = 1, NA
001212             WP = 0.0
001213             SP = 0.0
001214             IF (J .LT. I) SP = 1.0
001220             JJ = 1
001221        1217 KK = IDSP(JJ)
001223             IF (I .EQ. KK .AND. I .EQ. J) GO TO 2217
001232             IF (I .GT. KK) WP = WP + RIL(JJ,J,K)
001242             IF (KK .GT. I) GO TO 1216
001246             JJ = JJ + 1
001247             GO TO 1217
001250        1216 TIL(J,K) = SP - WP
001255             GO TO 1215
001256        2217 TIL(J,K) = +1.00
001262        1215 CONTINUE
001265        1214 CONTINUE
        C          LOAD THE LINE WITH THE LOAD SUBROUTINE
001267             CALL LOAD (TIL,TRLO,UNLO,SUBLEN,NA,IPT2,LENGTH,COOR,I,4HSHER,NS,
                  1IDSP,SPAN,STORE)
        C          PRINT THE SHEAR LOADINGS IF REQUIRED
001304             IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)
001323             GO TO 1220
        C          DETERMINE THE DESIGN SHEARS
```

```
001324      1220 DESSH(KJ,1) = STORE(1)                                          ANAL 274
001326           DESSH(KJ,3) = STORE(10)                                         ANAL 275
001330           PR = 0.0                                                        ANAL 276
001331           TR = 0.0                                                        ANAL 277
001332           DO 1230 KK = 2, 7                                               ANAL 278
001333           KB = 2 * (KK / 2 )                                              ANAL 279
001335           IF(ABS(STORE(KK)).LT..00001.AND.KB.EQ.KK) RR=STORE(8)+STORE(KK) ANAL 280
001351           IF(ABS(STORE(KK)).LT..00001.AND.KB.NE.KK) RR=STORE(9)+STORE(KK) ANAL 281
001366           IF (STORE(KK) .GT. 0.0) RR = STORE(9) + STORE(KK)               ANAL 282
001372           IF (STORE(KK) .LT. 0.0) RR = STORE(9) + STORE(KK)               ANAL 283
001376           IF (RR .GE. 0.0) GO TO 6230                                     ANAL 284
001400           IF (ABS(RR) .LT. TR) GO TO 1230                                 ANAL 285
001403           TR = ABS(RR)                                                    ANAL 286
001404           DESSH(KJ,4) = RR                                                ANAL 287
001405           GO TO 1230                                                      ANAL 288
001406      6230 IF (ABS(RR) .LT. PR) GO TO 1230                                 ANAL 289
001412           PR = ABS(RR)                                                    ANAL 290
001413           DESSH(KJ,2) = RR                                                ANAL 291
001414      1230 CONTINUE                                                        ANAL 292
001416           IF (KL .EQ. 0) GO TO 1201                                       ANAL 293
           C         CONSTRUCT THE LINE TO THE RIGHT OF THE POINT                ANAL 294
001417           KJ = 2 * I - 1                                                  ANAL 295
001421           KL = 0                                                          ANAL 296
001422           IF (MN .EQ. 1) GO TO 3100                                       ANAL 297
001423           DO 1250 J = 2, MN                                               ANAL 298
001425           IA = J                                                          ANAL 299
001426           IF (I .EQ. IOSP(J)) GO TO 1255                                  ANAL 300
001430      1250 CONTINUE                                                        ANAL 301
001432      3100 DO 1251 K = 1, 3                                                ANAL 302
001434      1251 TIL(I,K) = TIL(I,K) + 1.0                                       ANAL 303
           C         LOAD THE LINE WITH THE LOAD SUBROUTINE                      ANAL 304
001443           CALL LOAD(TIL,TRLO,INLO,SUBLEN,NA,IPT2,LENGTH,COOR,I,4HSHER,NS, ANAL 305
                1IOSP,SPAN,STORE)                                                ANAL 306
           C         PRINT THE SHEAR LOADINGS IF REQUIRED                        ANAL 307
001460           IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)   ANAL 308
001477           GO TO 1220                                                      ANAL 309
           C         CONSTRUCT THE LINE FOR THE RIGHT OF THE INTERIOR SUPPORTS   ANAL 310
001500      1255 DO 1260 J = 1, NA                                               ANAL 311
001502           SP = 0.0                                                        ANAL 312
001503           IF (J .LE. I) SP = 1.0                                          ANAL 313
001507           DO 1261 K = 1, 3                                                ANAL 314
001511           WP = 0.0                                                        ANAL 315
001512           DO 1262 KK = 1, IA                                              ANAL 316
001514      1262 WP = WP + RIL(KK,J,K)                                           ANAL 317
001524           IF (J .EQ. I) WP = 2.00                                         ANAL 318
001530      1261 TIL(J,K) = SP - WP                                              ANAL 319
001537      1260 CONTINUE                                                        ANAL 320
           C         LOAD THE LINE WITH THE LOAD SUBROUTINE                      ANAL 321
001541           CALL LOAD(TIL,TRLO,INLO,SUBLEN,NA,IPT2,LENGTH,COOR,I,4HSHER,NS, ANAL 322
                1IOSP,SPAN,STORE)                                                ANAL 323
           C         PRINT THE SHEAR LOADINGS IF REQUIRED                        ANAL 324
001557           IF (IPT10 .GE. 3) WRITE (6,1103) I, (STORE(JJJ), JJJ = 1, 10)   ANAL 325
001576           GO TO 1220                                                      ANAL 326
001577      1201 CONTINUE                                                        ANAL 327
           C         PRINT THE DESIGN SHEARS IF REQUIRED                         ANAL 328
001602           IF (IPT10 .EQ. 0) GO TO 102                                     ANAL 329
001603           WRITE (6,1231)                                                  ANAL 330
001606      1231 FORMAT (1H0,16HDESIGN SHEARS...//)                              ANAL 331
```

```
001606          KJ = 2 * NA - 2                                                  ANAL 332
001610          DO 1232 I = 1, KJ                                                ANAL 333
001612     1232 WRITE (6,1120) I,DESSH(I,1), DESSH(I,2), DESSH(I,3), DESSH(I,4)  ANAL 334
001632          IF (JPTS .GT. 9) WRITE (6,6010)                                  ANAL 335
001640      102 CONTINUE                                                         ANAL 336
001640          RETURN                                                           ANAL 337
001641          END                                                              ANAL 338

SUBPROGRAM LENGTH
002665

UNUSED COMPILER SPACE
034400



                FUNCTION BASFAT (S,FU,Y,YP)                                      BASF    1
          C         DETERMINES THE BASE METAL FATIGUE STRESS                     BASF    2
000007          DIMENSION S(4)                                                   BASF    3
          C         DETERMINE THE REQUIRED STRESS RATIO                          BASF    4
000007          SL = S(1) + S(2) + S(3)                                          BASF    5
000012          SR = S(1) + S(3) + S(4)                                          BASF    6
000015          R = SL / SR                                                      BASF    7
000017          IF (ABS(SL) .GT. ABS(SR)) R = SR / SL                            BASF    8
000026          SK = 1.0 + .78 * (FU/58. - 1.0)                                  BASF    9
000032          IF (Y .LT. 0.0) GO TO 700                                        BASF   10
          C         DETERMINE THE FATIGUE STRESS IN TENSION                      BASF   11
000034          SFA = SK * 20.5 / (1.0 - .55 * R)                                BASF   12
000040          GO TO 701                                                        BASF   13
          C         DETERMINE THE BASE METAL FATIGUE STRESS FOR COMPRESSION      BASF   14
000041      700 SFA = (.55*YP)/(1.0-((.55*YP/(SK*13.3))-1.0)*R)                  BASF   15
000050      701 BASFAT = SFA                                                     BASF   16
000052          RETURN                                                           BASF   17
000052          END                                                              BASF   18

SUBPROGRAM LENGTH
000113

UNUSED COMPILER SPACE
042700



                SUBROUTINE CONDES                                                COND    1
          C         DESIGNS THE SHEAR CONNECTOR SPACING FOR A GIVEN PROBLEM      COND    2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), COND    3
               1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),       COND    4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,    COND    5
               3SLABWD,SLABTH,MR,COOR(81),RTI(5,81,3),CC(5,5),FM,SPAN(4),IDSP(5), COND    6
               4TMOFI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLD(9),TPLO(10),PROP(3),COND    7
               5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),BEAR(5,3),COND    8
               6COST(9),CONSP(16,3,4)                                            COND    9
```

```
000002          COMMON/ONE/JPT3,JPT10
000002          COMMON/TWO/LPCM
000002          COMMON/THREE/AS,NCS,JPT6
000002          COMMON/TEN/HAUNCH
000002          REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH
000002          DIMENSION LPCM(10), SPACE(81), MMAX(4), C(16,3,4)
          C        ZERO OUT THE REQUIRED ARRAYS
000002          DO 2999 I = 1, 81
000004          SPACE(I) = 0.0
000005     2999 RO(I) = 0.0
000010          DO 4200 I = 1, 4
000011          DO 4200 J = 1, 16
000012          DO 4200 K = 1, 3
000013     4200 CONSP(J,K,I) = 0.0
          C        DESIGN THE SPACING FOR 7/8 INCH SHEAR CONNECTOR STUDS
000026          IF (JPT3 .LT. 9) GO TO 6000
000030          RETURN
000031     6000 DIA = .875
000033          ZR = 10.6 * DIA**2
000035          SLABA = SLABWD * SLABTH/MR
          C        DETERMINE THE NUMBER OF CONNECTORS AT EACH SECTION
000037          FMIN = 50.0
000041          DO 3000 I = 1, NF
000042          IA = NOS(I)
000044          IF (FLWID(IA) .LT. FMIN) FMIN = FLWID(IA)
000050     3000 CONTINUE
000053          NCS = 4
000054          IF (FMIN .LT. 12.0) NCS = 3
000057          IF (FMIN .GE. 15.0) NCS = 5
000063          SZR = NCS * ZR
          C        DETERMINE THE ACTUAL REQUIRED SPACING
000066          DO 3001 I = 1, NA
000067          DO 3010 J = 1, 10, 2
000070          IF (LPCM(J) .EQ. 0) GO TO 3011
000071          IF (I .GE. LPCM(J) .AND. I .LE. LPCM(J+1)) GO TO 3012
000102     3010 CONTINUE
000104     3011 IA = NOS(I)
000106          IC = 2 * I - 1
000110          ID = IC - 1
000111          IF (I .EQ. NA) GO TO 3002
000113          VR = DESSH(IC,2) - DESSH(IC,4)
000115          DD = DEPTH(IA) + PLATH(I,2) + HAUNCH + SLABTH
000122          SP = SLABA * (DD - CH(I) - SLABTH/2.)/IHCON(I)
000127          SPR = SZR/(VR * SP)
000131          IF (I .EQ. 1) GO TO 3003
000133     3002 VR = DESSH(ID,2) - DESSH(ID,4)
000136          IB = NOS(I-1)
000140          DD = DEPTH(IB) + PLATH(I-1,2) + HAUNCH + SLABTH
000144          SP = SLABA * (DD - CH(I-1) - SLABTH/2.)/IHCON(I-1)
000151          SPL = SZR/(VR * SP)
000154          IF (I .EQ. NA) GO TO 3004
000156          SPACE(I) = SPL
000157          IF (SPR .LT. SPL) SPACE(I) = SPR
000162          GO TO 3001
000163     3003 SPACE(I) = SPR
000165          GO TO 3001
000166     3004 SPACE(I) = SPL
000170          GO TO 3001
```

```
000171      3012 SPACE(I) = 0.0
000173      3001 CONTINUE
           C        DETERMINE REQUIRED SPACING TO THE NEAREST 3 INCH INCREMENT
000176           DO 3015 I = 1, NA
000177           N3 = SPACE(I) / 3.
000202           RO(I) = N3 * 3.
000205           IF (RO(I) .GT. 24.) RO(I) = 24.
000210      3015 CONTINUE
           C        FILL THE CONSP ARRAY WITH THE CORRECT VALUES
000213           MN = NS - 1
           C        BEGIN THE LOOP FOR EACH SPAN
000215           DO 3051 I = 1, MN
           C        DETERMINE THE CROSSOVER POINTS IN THE SPAN
000216           IF (MN .NE. 1) GO TO 3050
000220           IA = 1
000221           IB = NA
000222           GO TO 3053
000223      3050 IF (I .NE. 1) GO TO 3052
000225           IA = 1
000226           IB = LPCM(1) - 1
000230           GO TO 3053
000230      3052 IF (I .NE. MN) GO TO 3054
000232           DO 3055 JJ = 1, 10
000234           IA = LPCM(JJ) + 1
000236           IF (IA .GT. IOSP(MN)) GO TO 3056
000242      3055 CONTINUE
000244      3056 IB = NA
000246           GO TO 3053
000246      3054 DO 3057 J = 2, 10, 2
000250           IA = LPCM(J) + 1
000252           IB = LPCM(J+1) - 1
000254           IF (LPCM(J).GE.IOSP(I).AND.LPCM(J+1).LE.IOSP(I+1)) GO TO 3053
000265      3057 CONTINUE
           C        DETERMINE THE STARTING AND ENDING COORDINATE FOR THE CONNECTORS
000267      3053 MMAX(I) = IA
000271           DMAX = 0.0
000272           DO 4050 J = IA, IB
000273           IF (DESMOM(J,2) .LT. DMAX) GO TO 4050
000276           DMAX = DESMOM(J,2)
000277           MMAX(I) = J
000300      4050 CONTINUE
000303           SCOR = COOR(IA)
000305           FCOR = COOR(IA)
000306           J = 1
000307           DO 3080 K = IA, IB
000310           IF (K .EQ. IB) GO TO 3091
000312           IF (ABS(RO(K) - RO(K+1)) .LT. .00001) GO TO 3090
000316           IF (RO(K) .GT. RO(K+1)) GO TO 3085
000322           XO = (RO(K+1)-SPACE(K)) * SUBLEN(K)/(SPACE(K+1) - SPACE(K))
000326           NRB = XO / RO(K)
000331           XO = RO(K) * NRB + RO(K)
000334           FCOR = FCOR + XO
000335           GO TO 3089
000336      3085 XO = (SPACE(K)-RO(K)) * SUBLEN(K) / (SPACE(K) - SPACE(K+1))
000343           NRB = XO / RO(K)
000346           XO = RO(K) * NRB
000350           FCOR = FCOR + XO
000351      3089 CONSP(J,1,I) = RO(K)
```

```
000356              CONSP(J,2,I) = SCOR                                                                              COND 126
000361              CONSP(J,3,I) = FCOR                                                                              COND 127
000364              SCOR = FCOR                                                                                      COND 128
000365              FCOR = COOR(K+1)                                                                                 COND 129
000366              J = J + 1                                                                                        COND 130
000367              GO TO 3080                                                                                       COND 131
000370         3091 XD = COOR(IB) - SCOR                                                                             COND 132
000373              NRB = XD / RO(K)                                                                                 COND 133
000376              XD = XD - NRB * RO(K)                                                                            COND 134
000401              FCOR = COOR(IB) - XD                                                                             COND 135
000403              CONSP(J,1,I) = RO(K)                                                                             COND 136
000410              CONSP(J,2,I) = SCOR                                                                              COND 137
000413              CONSP(J,3,I) = FCOR                                                                              COND 138
000416              J = J + 1                                                                                        COND 139
000417              GO TO 3080                                                                                       COND 140
000420         3090 FCOR = FCOR + SUBLEN(K)                                                                          COND 141
000423         3080 CONTINUE                                                                                         COND 142
000426         3051 CONTINUE                                                                                         COND 143
              C           ADJUST THE SPACINGS FOR THE LAST SPAN                                                      COND 144
000430              DO 3100 J = 1, 16                                                                                COND 145
000432              JJ = J                                                                                           COND 146
000433              IF (ABS(CONSP(J,3,MN)) .LT. .0001) GO TO 3101                                                    COND 147
000441         3100 CONTINUE                                                                                         COND 148
000442         3101 JJ = JJ - 1                                                                                      COND 149
000444              XD = LENGTH - CONSP(JJ,3,MN)                                                                     COND 150
000450              DO 3102 J = 1, JJ                                                                                COND 151
000452              CONSP(J,3,MN) = CONSP(J,3,MN) + XD                                                               COND 152
000457         3102 CONSP(J,2,MN) = CONSP(J,2,MN) + XD                                                               COND 153
              C           CORRECT SPACING FOR INFLECTION POINTS                                                      COND 154
              C           DETERMINE NUMBER OF ROWS ADJACENT TO THE INFLECTION POINT                                  COND 155
000464              SUS = (.93 * DIA**2) * SQRT(2500. * FC)                                                          COND 156
000472              PSLAB = AS * 40.                                                                                 COND 157
000474              NSLAB = PSLAB/(.85 * SUS) + 1                                                                    COND 158
000501              NSER = AS * 10./78 + 1                                                                           COND 159
000506              IF (NSER .GT. NSLAB) NSLAB = NSER                                                                COND 160
000511              NRA = NSLAB/NCS + 1                                                                              COND 161
              C           CORRECT THE CONSP ARRAY                                                                    COND 162
000515              DO 4051 I = 1, MN                                                                                COND 163
000517              DO 4052 J = 1, 10                                                                                COND 164
000520              IF (LPCM(J) .EQ. 0) GO TO 4051                                                                   COND 165
000521              IA = LPCM(J)                                                                                     COND 166
000523              IF (IA .GT. IOSP(I) .AND. IA .LT. IOSP(I+1)) GO TO 4053                                          COND 167
000533              GO TO 4052                                                                                       COND 168
000533         4053 IB = IOSP(I)                                                                                     COND 169
000535              IF (COOR(IA) .GT. COOR(IB)+SPAN(I)/2.) GO TO 4054                                                COND 170
000544              KK = 2                                                                                           COND 171
000545              XD = DESMOM(IA+1,1)*SUBLEN(IA)/(ABS(DESMOM(IA+1,1)) +                                            COND 172
             -     1ABS(DESMOM(IA,1)))                                                                               COND 173
000552              GO TO 4155                                                                                       COND 174
000553         4054 KK = 3                                                                                           COND 175
000554              XD = DESMOM(IA-1,1)*SUBLEN(IA-1)/(ABS(DESMOM(IA-1,1)) +                                          COND 176
                   1ABS(DESMOM(IA,1)))                                                                               COND 177
000562         4155 IF (KK .EQ. 3) GO TO 4055                                                                        COND 178
000564              NXD = (CONSP(1,2,I) - COOR(IA+1) + XD) / CONSP(1,1,I)                                            COND 179
000575              XD = CONSP(1,1,I) * (NXD + NRA)                                                                  COND 180
000601              CONSP(1,KK,I) = CONSP(1,KK,I) - XD                                                               COND 181
000605              GO TO 4052                                                                                       COND 182
000605         4055 DO 4056 JA = 1, 16                                                                               COND 183
```

```
000607         K = JK - 1                                                          COND 184
000611         IF (ABS(CONSP(JK,1,I)) .LT. .00001) GO TO 4057                      COND 185
000616    4056 CONTINUE                                                            COND 186
000620    4057 NXD = (COOR(IA-1) + XD - CONSP(K,KK,I)) / CONSP(K,1,I)             COND 187
000633         XD = CONSP(K,1,I) * (NXD + NPA)                                     COND 188
000640         CONSP(K,KK,I) = CONSP(K,3,I) + XD                                   COND 189
000645    4052 CONTINUE                                                            COND 190
000647    4051 CONTINUE                                                            COND 191
         C          CORRECTION FOR ULTIMATE STRENGTH                               COND 192
000652         DO 4080 I = 1, MN                                                   COND 193
000653         IA = MMAX(I)                                                        COND 194
000655         IB = NOS(IA)                                                        COND 195
000657         AR = SAREA(IB) + PLATH(IA,1)*PLWID(IA,1) + PLWID(IA,2)*PLATH(IA,2)COND 196
000664         PGOV = AR * PROP(3)                                                 COND 197
000666         PCONC = 2.125 * FC * SLABWD * SLABTH                                COND 198
000671         IF (PCONC .LT. PGOV) PGOV = PCONC                                   COND 199
000674         NREQ = PGOV / (.85 * SUS) + 1                                       COND 200
000701         DO 4081 K = 1, 16                                                   COND 201
000703         IF (ABS(CONSP(K,1,I)) .LT. .0001) GO TO 4082                        COND 202
000711         IF(COOR(IA).LE.CONSP(K,3,I).AND.COOR(IA).GE.CONSP(K,2,I)) JK = K   COND 203
000727         KJ = K                                                              COND 204
000731    4081 CONTINUE                                                            COND 205
000732    4082 ROW = 1.0                                                           COND 206
000734         J = 1                                                               COND 207
000735    4084 IF (J .EQ. JK) GO TO 4083                                           COND 208
000737         ROW = ROW + (CONSP(J,3,I) - CONSP(J,2,I)) / CONSP(J,1,I)            COND 209
000747         J = J + 1                                                           COND 210
000750         GO TO 4084                                                          COND 211
000751    4083 ROW = ROW + (COOR(IA) - CONSP(J,2,I)) / CONSP(J,1,I)                COND 212
000762         NCON = ROW * NCS + 1                                                COND 213
000766         IF (ABS(NCON - ROW * NCS) .GT. .00001) NCON = NCON - 1              COND 214
000775         IF (NCON .GE. NREQ) GO TO 4086                                      COND 215
001000         NADD = (NREQ - NCON) / NCS + 1                                      COND 216
001004         NAF = NADD                                                          COND 217
001005         NTOT = 0                                                            COND 218
001006         NT = 0                                                              COND 219
001007         KKK = JK - 1                                                        COND 220
001010         IF (KKK .EQ. 0) GO TO 180                                           COND 221
001011         NAF = NADD / KKK                                                    COND 222
001013         DO 4550 J = 1, KKK                                                  COND 223
001015         NTOT = NTOT + NAF                                                   COND 224
001017         IF (J .EQ. KKK .AND. NTOT .NE. NADD) NAF=NADD-NTOT+NAF              COND 225
001027         DM = CONSP(J,3,I) - CONSP(J,2,I)                                    COND 226
001036         NSP = DM / CONSP(J,1,I) + NAF                                       COND 227
001042         SP = DM / NSP                                                       COND 228
001044         NP3 = SP / 3.                                                       COND 229
001047         SP = NP3 * 3.                                                       COND 230
001051         CONSP(J,1,I) = SP                                                   COND 231
001053         NT = NT + DM / SP - NSP + NAF                                       COND 232
001062         IF (NT .GE. NADD) GO TO 4086                                        COND 233
001064    4550 CONTINUE                                                            COND 234
001067         GO TO 4086                                                          COND 235
001067     180 D = (COOR(IA) - CONSP(JK,2,I)                                       COND 236
001074         NSP = D / CONSP(JK,1,I) + NAF                                       COND 237
001102         SP = D / NSP                                                        COND 238
001104         NP3 = SP / 3.                                                       COND 239
001107         SP = NP3 * 3.                                                       COND 240
001111     185 KB = 15                                                             COND 241
```

```
001112        TEMP = CONSP(1,2,I)                                                  COND 242
001116    188 DO 186 KC = 1, 3                                                     COND 243
001120    186 CONSP(KB+1,KC,I) = CONSP(KB,KC,I)                                    COND 244
001134        IF (KB .EQ. 1) GO TO 187                                             COND 245
001136        KB = KB - 1                                                          COND 246
001137        GO TO 188                                                            COND 247
001137    187 CONSP(1,1,I) = SP                                                    COND 248
001143        CONSP(1,2,I) = TEMP                                                  COND 249
001146        CONSP(1,3,I) = CONSP(1,2,I) + SP * NSP                               COND 250
001154        CONSP(2,2,I) = CONSP(1,3,I)                                          COND 251
001157   4086 ROW = 1.0                                                            COND 252
001161        J = KJ                                                               COND 253
001162   4089 IF (J .EQ. JK) GO TO 4090                                            COND 254
001164        ROW = ROW + (CONSP(J,3,I) - CONSP(J,2,I)) / CONSP(J,1,I)             COND 255
001174        J = J - 1                                                            COND 256
001175        GO TO 4089                                                           COND 257
001176   4090 ROW = ROW + (CONSP(J,3,I) - COOR(IA)) / CONSP(J,1,I)                 COND 258
001207        NCON = ROW * NCS + 1                                                 COND 259
001214        IF (ABS(NCON - ROW * NCS) .GT. .0001) NCON = NCON - 1                COND 260
001223        IF (NCON .GE. NREQ) GO TO 4500                                       COND 261
001226        NADD = (NREQ - NCON) / NCS + 1                                       COND 262
001232        NAF = NADD                                                           COND 263
001233        IF (KJ .EQ. JK) GO TO 150                                            COND 264
001235        KKK = KJ - JK                                                        COND 265
001236        NAF = NADD / KKK                                                     COND 266
001240        J = KJ                                                               COND 267
001241        NTOT = 0                                                             COND 268
001242        NT = 0                                                               COND 269
001243    151 NTOT = NTOT + NAF                                                    COND 270
001245        IF (J.EQ.JK+1 .AND. NTOT.NE. NADD) NAF=NADD-NTOT+NAF                 COND 271
001257        DM = CONSP(J,3,I) - CONSP(J,2,I)                                     COND 272
001266        NSP = DM / CONSP(J,1,I) + NAF                                        COND 273
001272        SP = DM / NSP                                                        COND 274
001274        NP3 = SP / 3.                                                        COND 275
001277        SP = NP3 * 3.                                                        COND 276
001301        CONSP(J,1,I) = SP                                                    COND 277
001303        NT = NT + DM/SP - NSP + NAF                                          COND 278
001312        IF (NT .GE. NADD) GO TO 4080                                         COND 279
001314        J = J - 1                                                            COND 280
001316        IF (J .EQ. JK) GO TO 4080                                            COND 281
001317        GO TO 151                                                            COND 282
001320    150 D = CONSP(JK,3,I) - COOR(IA)                                         COND 283
001325        NSP = D / CONSP(JK,1,I) + NAF                                        COND 284
001333        SP = D / NSP                                                         COND 285
001335        NP3 = SP / 3.                                                        COND 286
001340        SP = NP3 * 3.                                                        COND 287
001342        CONSP(JK+1,1,I) = SP                                                 COND 288
001346        CONSP(JK+1,3,I) = CONSP(JK,3,I)                                      COND 289
001352        CONSP(JK+1,2,I) = CONSP(JK+1,3,I) - NSP * SP                         COND 290
001356        CONSP(JK,3,I) = CONSP(JK+1,2,I)                                      COND 291
001361   4500 CONTINUE                                                             COND 292
001361   4080 CONTINUE                                                             COND 293
001364        DO 40 I = 1, NN                                                      COND 294
001365        KSCI = 1                                                             COND 295
001366        DO 41 J = 1, 16                                                      COND 296
001370        IF (ABS(CONSP(J,1,I)) .LT. .00001) GO TO 39                          COND 297
001376        D = CONSP(J,3,I) - CONSP(J,2,I)                                      COND 298
001403        NKD = D / CONSP(J,1,I)                                               COND 299
```

```
001406         IF (ABS(NXD*CONSP(J,1,I) - D) .GT. .0001) NXD = NXD + 1
001415         IF (NXD .GT. 3) GO TO 41
001421         KSCT = 1
001422         IF (J .EQ. 1) GO TO 42
001424         IF (ABS(CONSP(J+1,1,I)) .LT. .0001) GO TO 43
001430         CONSP(J,1,I) = CONSP(J-1,1,I)
001433         IF (CONSP(J+1,1,I).LT.CONSP(J-1,1,I)) CONSP(J,1,I)=CONSP(J+1,1,I)
001440         GO TO 41
001441      42 CONSP(1,3,I) = CONSP(1,2,I) + 3. * CONSP(1,1,I)
001450         CONSP(2,2,I) = CONSP(1,3,I)
001453         GO TO 41
001453      43 CONSP(J,2,I) = CONSP(J,3,I) - 3. * CONSP(J,1,I)
001463         CONSP(J-1,3,I) = CONSP(J,2,I)
001466      41 CONTINUE
001470      39 IF (KSCT .EQ. 0) GO TO 40
001471         DO 45 J = 1, 16
001473         DO 45 K = 1, 3
001474         C(J,K,I) = CONSP(J,K,I)
001504      45 CONSP(J,K,I) = 0.0
001512         KB = 1
001513         DO 46 J = 1, 16
001515         IF (ABS(C(J,1,I)) .LT. .0001) GO TO 40
001523         IF (J .EQ. 1) GO TO 47
001524         IF (ABS(C(J-1,1,I) - C(J,1,I)) .GT. .0001) GO TO 47
001535         CONSP(KB-1,3,I) = C(J,3,I)
001541         GO TO 46
001541      47 CONSP(KB,1,I) = C(J,1,I)
001547         CONSP(KB,2,I) = C(J,2,I)
001552         CONSP(KB,3,I) = C(J,3,I)
001555         KB = KB + 1
001556      46 CONTINUE
001560      40 CONTINUE
         C          FINAL ADJUSTMENTS OF THE CONNECTOR SPACING
001563         DO 4390 I = 1, MN
001564         DO 4390 J = 1, 16
001565         IF (ABS(CONSP(J,1,I)) .LT. .0001) GO TO 4390
001573         XD = CONSP(J,3,I) - CONSP(J,2,I)
001600         NXD = XD / CONSP(J,1,I)
001603         IF (ABS(NXD*CONSP(J,1,I) - XD) .GT. .0001) NXD = NXD + 1
001612         XD = NXD * CONSP(J,1,I)
001616         IF (CONSP(J+1,1,I) .LT. CONSP(J,1,I)) GO TO 4092
001622         CONSP(J,3,I) = CONSP(J,2,I) + XD
001625         CONSP(J+1,2,I) = CONSP(J,3,I)
001627         GO TO 4390
001627    4092 IF(ABS(CONSP(J+1,1,I)) .LT. .0001) GO TO 4093
001635         CONSP(J,3,I) = CONSP(J,2,I) + XD - CONSP(J,1,I)
001644         CONSP(J+1,2,I) = CONSP(J,3,I)
001647         GO TO 4390
001647    4093 CONSP(J,3,I) = CONSP(J,2,I) + XD
001656    4390 CONTINUE
001663         IF (MN .EQ. 1) GO TO 11
001665         DO 4095 J = 1, 16
001666         JJ = J
001667         IF (ABS(CONSP(J,3,MN)) .LT. .0001) GO TO 4096
001675    4095 CONTINUE
001676    4096 JJ = JJ - 1
001700         XD = LENGTH - CONSP(JJ,3,MN)
001704         DO 4097 J = 1, JJ
```

```
001706          CONSP(J,3,MN) = CONSP(J,3,MN) + XD                            COND 358
001713     4097 CONSP(J,2,MN) = CONSP(J,2,MN) + XD                            COND 359
001720       10 CONTINUE                                                      COND 360
001720          RETURN                                                        COND 361
001721          END                                                           COND 362

SUBPROGRAM LENGTH
002531

UNUSED COMPILER SPACE
034100


                SUBROUTINE CPDES                                              CPDS    1
         C         DESIGN THE COVER PLATE ARRANGEMENTS FOR A ROLLED SECTION   CPDS    2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), CPDS    3
               1DAF(18),IX(18),IY(18),USECT(18),NOS(80),ISTEEL(80),CS(80),    CPDS    4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF, CPDS    5
               3SLABWD,SLABTH,MR,COOR(81),RI(5,81,3),CC(5,5),FM,SPAN(4),IOSP(5), CPDS    6
               4TMDEI(81),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLD(9),TRLD(10),PROP(3),CPDS    7
               5EC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),REAR(5,3),CPDS    8
               6COST(9),CONSP(16,3,4),COVPL(12,6,2)                           CPDS    9
000002          COMMON/ONE/IPT3,IPT10                                         CPDS   10
000002          COMMON/TWO/LPCM                                               CPDS   11
000002          COMMON/SIX/FU                                                 CPDS   12
000002          COMMON/EIGHT/IST                                              CPDS   13
000002          COMMON/TEN/HAUNCH                                             CPDS   14
000002          COMMON/ELEVEN/KD,KC,SIS,SHS,SLS,ISS,ILSS,DMAX,DIST            CPDS   15
000002          COMMON/FEEN/PTH,KT,I                                          CPDS   16
000002          REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                       CPDS   17
000002          DIMENSION LPCM(10),PTH(40,4),PSH(40,4),PHH(40,4)              CPDS   18
000002          IF (IPT10 .GE. 3) WRITE (6,1040)                              CPDS   19
000011     1040 FORMAT (1H1,20X,11HSTART CPDES//)                             CPDS   20
         C         ZERO OUT THE REQUIRED ARRAYS                               CPDS   21
000011          DO 9400 I = 1, 12                                             CPDS   22
000013          DO 9400 J = 1, 6                                              CPDS   23
000014          DO 9400 K = 1, 2                                              CPDS   24
000015     9400 COVPL(I,J,K) = 0.0                                            CPDS   25
000031          MN = NS - 1                                                   CPDS   26
000033          J = 2                                                         CPDS   27
000034     6300 DO 9001 I = 1, NF                                             CPDS   28
000036          DO 9001 JJ = 1, 4                                             CPDS   29
000037     9001 PTH(I,JJ) = 0.0                                               CPDS   30
000047          DO 5061 I = 1, 40                                             CPDS   31
000051          DO 5061 K = 1, 4                                              CPDS   32
000052     5061 PHH(I,K) = 0.0                                                CPDS   33
000062          KB = 1                                                        CPDS   34
000063          NC = KB                                                       CPDS   35
         C         DETERMINE THE PTH ARRAY                                    CPDS   36
000064          DO 9600 I = 1, NF                                             CPDS   37
000065          IF (I .GT. 1) GO TO 9500                                      CPDS   38
000070     9510 PTH(KB,1) = PLATH(I,1)                                        CPDS   39
000075          PTH(KB,2) = PLWID(I,1)                                        CPDS   40
000100          PTH(KB,3) = COOR(I)                                           CPDS   41
```

```
000101            PTH(KB,4) = COOR(I+1)                                            CPDS  42
000103            GO TO 9000                                                       CPDS  43
000103      9500 IF (ABS(PLATH(I,J)-PTH(KB,1)).LT..00001) GO TO 9600               CPDS  44
000113            KB = KB + 1                                                      CPDS  45
000114            NC = KB                                                          CPDS  46
000115            GO TO 9510                                                       CPDS  47
000115      9600 PTH(KB,4) = COOR(I+1)                                             CPDS  48
000120      9000 CONTINUE                                                          CPDS  49
000123            KSST = 0                                                         CPDS  50
000124            IF (MN .EQ. 1) GO TO 7002                                        CPDS  51
000126            IF (J .EQ. 1) GO TO 7002                                         CPDS  52
000127            DO 7001 II = 2, NS                                               CPDS  53
000131            I = IOSP(II)                                                     CPDS  54
000133            DO 7003 JJ = 1, NC                                               CPDS  55
000134            IF (COOR(I).GT.PTH(JJ,3).AND.COOR(I).LT.PTH(JJ,4)) GO TO 7005    CPDS  56
000146            GO TO 7003                                                       CPDS  57
000146      7005 IQ = JJ                                                           CPDS  58
000150            IF (PTH(IQ-1,1) .LT. .1) GO TO 7006                              CPDS  59
000153            KSST = 1                                                         CPDS  60
000154            PTH(IQ-1,1) = 0.0                                                CPDS  61
000155            PTH(IQ-1,2) = 0.0                                                CPDS  62
000156      7006 IF (PTH(IQ+1,1) .LT. .1) GO TO 7003                               CPDS  63
000161            KSST = 1                                                         CPDS  64
000162            PTH(IQ+1,1) = 0.0                                                CPDS  65
000163            PTH(IQ+1,2) = 0.0                                                CPDS  66
000164      7003 CONTINUE                                                          CPDS  67
000167      7001 CONTINUE                                                          CPDS  68
000171            IF (KSST .EQ. 0) GO TO 7002                                      CPDS  69
000172            KB = NC                                                          CPDS  70
000174            KTT = 1                                                          CPDS  71
000175            NC = 1                                                           CPDS  72
000176            DO 7010 I = 1, KB                                                CPDS  73
000177            IF (I .GT. 1) GO TO 7015                                         CPDS  74
000202      7020 DO 7012 KKK = 1, 4                                                CPDS  75
000204      7012 PTH(KTT,KKK) = PTH(I,KKK)                                         CPDS  76
000215            GO TO 7010                                                       CPDS  77
000215      7015 IF (ABS(PTH(I,1) - PTH(I-1,1)) .LT. .0001) GO TO 7016             CPDS  78
000222            KTT = KTT + 1                                                    CPDS  79
000224            NC = KTT                                                         CPDS  80
000225            GO TO 7020                                                       CPDS  81
000225      7016 PTH(KTT,4) = PTH(I,4)                                             CPDS  82
000230      7010 CONTINUE                                                          CPDS  83
           C         PRINT THE ORIGINAL PTH ARRAY IF REQUIRED                      CPDS  84
000233      7002 IF (IPTIQ .LT. 3) GO TO 100                                       CPDS  85
000236            WRITE (6,1000)                                                   CPDS  86
000241      1000 FORMAT (1H0,10X,18HORIGINAL PTH ARRAY//)                          CPDS  87
000241            DO 1001 I = 1, NC                                                CPDS  88
000243      1001 WRITE (6,1002) (PTH(I,K), K = 1, 4)                               CPDS  89
000261      1002 FORMAT (2X,9HPLATE IS ,F10.5,3H X ,F6.2,10X,8HSTART = ,F10.2,10X, CPDS  90
                 16HEND = ,F10.2)                                                  CPDS  91
000261       100 KB = 0                                                            CPDS  92
000262            KILL = 0                                                         CPDS  93
000263            KPST = 0                                                         CPDS  94
000264            IF (NC .EQ. 1 .AND. ABS(PTH(1,1)) .LT. .0001) GO TO 102          CPDS  95
           C         START THE LOOP FOR THE PLATE CUTOFF                           CPDS  96
000276            DO 9300 I = 1, NC                                                CPDS  97
000277            KST = KB                                                         CPDS  98
000301            K9 = 0                                                           CPDS  99
```

```
000301           KT = 0                                                              CPDS 100
000302           IF (IPTIP .GE. 3) WRITE (6,1041) I                                  CPDS 101
000313      1041 FORMAT (/5X,21HCALCULATIONS FOR I = ,I2)                            CPDS 102
000313           IF (PTH(I,1) .LT. .1) GO TO 9300                                    CPDS 103
000316           IF (NILL .EQ. 0) GO TO 190                                          CPDS 104
000317           IF (ABS(PTH(I,1)-PTH(I-1,1)) .GT. .0001) GO TO 190                  CPDS 105
000325           GO TO 9300                                                          CPDS 106
000325       190 IF(ABS(PTH(I,1)-PTH(I-1,1)) .LT. .0001) GO TO 9300                  CPDS 107
000332           IF (KST .EQ. 2) GO TO 9359                                          CPDS 108
          C         DETERMINE THE PLATE END CONDITIONS                               CPDS 109
000334           DO 9330 K = 1, NA                                                   CPDS 110
000336           IA = K                                                              CPDS 111
000337           IF (ABS(COOR(IA)-PTH(I,3)).LT..00001) GO TO 9331                    CPDS 112
000345      9330 CONTINUE                                                            CPDS 113
000347      9331 KO = IA                                                             CPDS 114
000351           KC = 1                                                              CPDS 115
000352           IF (I .EQ. 1) GO TO 9333                                            CPDS 116
000354           IF (PTH(I-1,1) .GT. .1) KC = 2                                      CPDS 117
000357      9333 ISS = 1                                                             CPDS 118
000360           IF (PTH(I-1,1) .GT. PTH(I,1)) ISS = 2                               CPDS 119
000364           IF (IPTIP .EQ. 4) WRITE (6,1012) KO, KC, ISS                        CPDS 120
000400      1012 FORMAT (/5X,5HKO = ,I2,2X,5HKC = ,I2,2X,6HISS = ,I2)                CPDS 121
000400           IF (MN .EQ. 1) GO TO 9329                                           CPDS 122
          C         DETERMINE THE MAXIMUM ALLOWABLE CUTOFF                           CPDS 123
000402           DO 9310 K = 2, MN                                                   CPDS 124
000404           ID = IOSP(K)                                                        CPDS 125
000406           IF (KPST .EQ. 1) GO TO 7700                                         CPDS 126
000410           IF (COOR(ID).GE.PTH(I,3) .AND. COOR(ID).LE.PTH(I,4)) GO TO 9315     CPDS 127
000422           GO TO 9310                                                          CPDS 128
000422      7700 IF (COOR(ID) .GE. PTH(I-1,3) .AND. COOR(ID) .LE. PTH(I-1,4))        CPDS 129
                1GO TO 9317                                                          CPDS 130
000434      9310 CONTINUE                                                            CPDS 131
000437      9329 IF (NILL .EQ. 1) GO TO 9317                                         CPDS 132
000441           DMAX = SUBLEN(IA)                                                   CPDS 133
000443           IF (ISS .EQ. 2) DMAX = SUBLEN(IA-1)                                 CPDS 134
000447           GO TO 9321                                                          CPDS 135
000450      9315 IF (KR .EQ. 2) GO TO 9317                                           CPDS 136
000452           NDD = DEPTH(IST) + 19.                                              CPDS 137
000456           D = NDD * 1.0                                                       CPDS 138
000460           DMAX = COOR(ID) - D - PTH(I,3)                                      CPDS 139
000464           KR = 1                                                              CPDS 140
000465           GO TO 9321                                                          CPDS 141
000465      9317 NDD = DEPTH(IST) + 19.                                              CPDS 142
000471           D = NDD * 1.0                                                       CPDS 143
000473           DMAX = PTH(I,4) - COOR(ID) - D                                      CPDS 144
000477           KPST = 0                                                            CPDS 145
000500           IF (NILL .EQ. 1) DMAX = PTH(I,3) - COOR(ID) - D                     CPDS 146
000506      9321 KK = KO                                                             CPDS 147
000510           IF (IPTIP .EQ. 4) WRITE (6,1026) DMAX                               CPDS 148
000517      1026 FORMAT (6X,7HDMAX = ,F15.2)                                         CPDS 149
000517           IF (NILL .EQ. 1) GO TO 290                                          CPDS 150
000521           IF (ISTEEL(KO) .GT. ISTEEL(KO-1)) KK = KO -                         CPDS 151
000525           IF (J .EQ. 2) GO TO 6024                                            CPDS 152
          C         DETERMINE THE SECTION PROPERTIES AT THE PLATE END                CPDS 153
000527           TD = DEPTH(IST) + PLATH(KK,1) + PLATH(KK,2)                         CPDS 154
000533           GO TO 6025                                                          CPDS 155
000534      6024 TD = 0.                                                             CPDS 156
000535      6025 SIS = ISTEEL(KK) / ABS(TD - CS(KK))                                 CPDS 157
```

```
000542          PLT = PLATH(KK,2)                                                  CPDS 158
000543          PLW = PLWID(KK,2)                                                  CPDS 159
000545          IF (JPTI .GT. 9) GO TO 490                                         CPDS 160
000550          SLS = ILCON(KK) / ABS(TD - CL(KK))                                 CPDS 161
000554          SHS = IHCON(KK) / ABS(TD - CH(KK))                                 CPDS 162
000560      490 APROD = PLT*PLW*(DEPTH(IST)/2.+PLT/2.)**2                          CPDS 163
000565          AISEC = (PLW * PLT**3) / 12.                                       CPDS 164
000570          SS = IX(IST) + 2.*AISEC + 2.*APROD                                 CPDS 165
000575          SS = SS / (DEPTH(IST)/2. + PLT)                                    CPDS 166
000600          IF (JPTI .GT. 9) SIS = SS                                          CPDS 167
000604          IF (JPTI .GT. 9) SLS = SS                                          CPDS 168
000607          IF (JPTI .GT. 9) SHS = SS                                          CPDS 169
000612          IF (KILL .EQ. 1) GO TO 590                                         CPDS 170
000614          IF (ABS(PLATH(KK,1)).LT..0001 .AND. J .EQ. 1) SS = SIS             CPDS 171
000630          GO TO 390                                                          CPDS 172
000631      290 PLT = PTH(I-1,1)                                                   CPDS 173
000633          PLW = PTH(I-1,2)                                                   CPDS 174
000635          IF (PTH(I,1) .LT. PTH(I-1,1)) PLT = PTH(I,1)                       CPDS 175
000641          IF (PTH(I,1) .LT. PTH(I-1,1)) PLW = PTH(I,2)                       CPDS 176
000645          IF (JPTI .GT. 9) GO TO 490                                         CPDS 177
000652          TD = DEPTH(IST) + PLT                                              CPDS 178
000654          IF (J .EQ. 2) TD = 0.0                                             CPDS 179
000657          HP = DEPTH(IST)/2. + PLT                                           CPDS 180
000663          CALL ICALC(0.,0.,0.,0.,PLW,PLT,DEPTH(IST),SAREA(IST),IX(IST),HP,0.,  CPDS 181
               1SC,SIS)                                                            CPDS 182
000677          TSA = SAREA(IST) + PLT*PLW                                         CPDS 183
000703          H = 2. * (DEPTH(IST) + PLT - SC)                                   CPDS 184
000706          CALL ICALC(SLABWO/(3.*MR),SLABTH,0.,0.,H,TSA,SIS,SC,HAUNCH,CSL,SL)CPDS 185
000723          CALL ICALC(SLABWO/MR,SLABTH,0.,0.,H,TSA,SIS,SC,HAUNCH,CSH,SH)      CPDS 186
000740          SIS = SIS/ABS(TD - SC)                                             CPDS 187
000744          SHS = SH/ABS(TD - CSH)                                             CPDS 188
000747          SLS = SL/ABS(TD - CSL)                                             CPDS 189
000753          GO TO 490                                                          CPDS 190
000753      590 CONTINUE                                                           CPDS 191
000753      390 KILL = 0                                                           CPDS 192
000754          IF (IPT10 .EQ. 4) WRITE (6,1027) SIS, SLS, SHS, SS                 CPDS 193
000772     1027 FORMAT (2X,6HSIS = ,F10.3,2X,6HSLS = ,F10.3,2X,6HSHS = ,F10.3,2X,  CPDS 194
               15HSS = ,F10.3)                                                     CPDS 195
        C           CALL ON THE CUTOFF SUBROUTINE                                  CPDS 196
000772          CALL TERMIN                                                        CPDS 197
        C           PRINT THE CUTOFF DISTANCE IF REQUIRED                          CPDS 198
000773          IF (IPT10 .GE. 3) WRITE (6,110) DIST                               CPDS 199
001004      110 FORMAT (/20X,18HCUTOFF DISTANCE = ,F15.5//)                        CPDS 200
        C           ADJUST THE PTH ARRAY                                           CPDS 201
001004          IF (KT .EQ. 1) GO TO 9360                                          CPDS 202
001006          PTH(I-1,4) = PTH(I-1,4) + DIST                                     CPDS 203
001011          PTH(I,3) = PTH(I-1,4)                                              CPDS 204
001012     7000 IF (IPT10 .EQ. 4) WRITE (6,1002) (PTH(I-1,K), K = 1, 4)            CPDS 205
001030          IF (IPT10 .EQ. 4) WRITE (6,1002) (PTH(I,K), K = 1, 4)              CPDS 206
001046          IF (KR .GT. 1) GO TO 9358                                          CPDS 207
001051          GO TO 9359                                                         CPDS 208
001051     9360 PTH(I,4) = PTH(I,4) + DIST                                         CPDS 209
001054          IF (I .EQ. NC) GO TO 9300                                          CPDS 210
001056          PTH(I+1,3) = PTH(I,4)                                              CPDS 211
001057          IF (IPT10 .EQ. 4) WRITE (6,1002) (PTH(I,K), K = 1, 4)              CPDS 212
001075          IF (IPT10 .EQ. 4) WRITE (6,1002) (PTH(I+1,K), K = 1, 4)            CPDS 213
001113          IF (ABS(PTH(I+1,4)-PTH(I+1,3)) .GT. .0001) GO TO 9300              CPDS 214
001121          KR = 0                                                             CPDS 215
```

```
001121           KILL = 1                                                          CPDS 216
001122           IF (PTH(I+2,1) .LT. PTH(I,1)) GO TO 94                            CPDS 217
001125           PTH(I+1,1) = PTH(I,1)                                             CPDS 218
001126           PTH(I+1,2) = PTH(I,2)                                             CPDS 219
001130           GO TO 9300                                                        CPDS 220
001130        94 PTH(I+1,1) = PTH(I+2,1)                                           CPDS 221
001132           PTH(I+1,2) = PTH(I+2,2)                                           CPDS 222
001134           PTH(I+1,3) = PTH(I+1,3) - DIST                                    CPDS 223
001136           PTH(I,4) = PTH(I+1,3)                                             CPDS 224
001137           KPST = 1                                                          CPDS 225
001140           GO TO 9300                                                        CPDS 226
001141      9359 IF (I .EQ. NC) GO TO 9368                                         CPDS 227
001143           IF (PTH(I+1,1) .GT. .1) GO TO 9300                                CPDS 228
001147           IF (KR .LT. .5) GO TO 9368                                        CPDS 229
001152      9358 KR = 2                                                            CPDS 230
001153      9368 KT = 1                                                            CPDS 231
           C           CUTOFF THE RIGHT END IF REQUIRED                            CPDS 232
001154           DO 9357 K = 1, NA                                                 CPDS 233
001156           IH = K                                                            CPDS 234
001157           IF (ABS(COOR(IH)-PTH(I,4)).LT..00001) GO TO 9356                  CPDS 235
001165      9357 CONTINUE                                                          CPDS 236
001167      9356 KQ = IH                                                           CPDS 237
001171           KC = 1                                                            CPDS 238
001172           ISS = 2                                                           CPDS 239
001173           IF (IPT10 .EQ. 4) WRITE (6,1012) KQ, KC, ISS                      CPDS 240
001206           IF (KR .EQ. 2) GO TO 9371                                         CPDS 241
001210           GO TO 9329                                                        CPDS 242
001211      9371 IF (PTH(I+1,1) .GT. .1) KC = 2                                    CPDS 243
001215           IF (PTH(I,1) .LT. PTH(I+1,1)) ISS = 1                             CPDS 244
001221           IF (IPT10 .EQ. 4) WRITE (6,1012) KQ, KC, ISS                      CPDS 245
001235           GO TO 9315                                                        CPDS 246
001236      9300 CONTINUE                                                          CPDS 247
001241           NTOT = NC                                                         CPDS 248
           C           SET UP THE PHH ARRAY                                        CPDS 249
001242           DO 5060 I = 1, NTOT                                               CPDS 250
001243           DO 5060 K = 1, 4                                                  CPDS 251
001244      5060 PHH(I,K) = PTH(I,K)                                               CPDS 252
001257           KIPPER = 0                                                        CPDS 253
001260           NDOP = 37. + 2. * DEPTH(IST)                                      CPDS 254
001264           DOP = NDOP * 1.                                                   CPDS 255
           C           SET UP THE OPTIMIZATION PROCEDURE IF END PLATES ARE TOO SHORT CPDS 256
001267           IZS = 1                                                           CPDS 257
001270           IZF = NTOT                                                        CPDS 258
001271           IF (PHH(1,1) .LT. .1) IZS = 2                                     CPDS 259
001274           IF (PHH(NTOT,1) .LT. .1) IZF = NTOT - 1                           CPDS 260
001301           DS = PHH(IZS,4) - PHH(IZS,3)                                      CPDS 261
001304           DF = PHH(IZF,4) - PHH(IZF,3)                                      CPDS 262
001306           IF (DS .GT. DOP .AND. DF .GT. DOP) GO TO 5025                     CPDS 263
001317           FTOT = 999999999.                                                 CPDS 264
001320           KIPS = 0                                                          CPDS 265
001321           IF (DS .GE. DOP) GO TO 5080                                       CPDS 266
001323           KIPS = 1                                                          CPDS 267
001324           IF (IZS .EQ. 1) GO TO 5080                                        CPDS 268
001326           IF (DOP - DS .GT. PHH(1,4) - PHH(1,3)) GO TO 5080                 CPDS 269
001333           KIPS = 2                                                          CPDS 270
001334      5080 KIPF = 0                                                          CPDS 271
001335           IF (DF .GE. DOP) GO TO 5081                                       CPDS 272
001340           KIPF = 1                                                          CPDS 273
```

```
001341          IF (IZF .EQ. NTOT) GO TO 5081                                   CPDS 274
001343          IF (LOP - DF .LT. PHH(NTOT,4) - PHH(NTOT,3)) GO TO 5081         CPDS 275
001350          KIPF = 2                                                        CPDS 276
001351     5081 KIPPER = 1                                                      CPDS 277
001352          KPS = KIPS                                                      CPDS 278
001354          KPF = KIPF                                                      CPDS 279
001355     5500 IF (KPS .EQ. 0) GO TO 55                                        CPDS 280
001356          IF (KPS .EQ. 1) GO TO 54                                        CPDS 281
001360          PTH(IZS,3) = PTH(IZS,4) - DOP                                   CPDS 282
001363          PTH(IZS-1,4) = PTH(IZS,3)                                       CPDS 283
001364          GO TO 55                                                        CPDS 284
001365       54 PTH(IZS,1) = PTH(IZS-1,1)                                       CPDS 285
001367       55 IF (KPF .EQ. 0) GO TO 5025                                      CPDS 286
001370          IF (KPF .EQ. 1) GO TO 56                                        CPDS 287
001372          PTH(IZF,4) = PTH(IZF,3) + DOP                                   CPDS 288
001375          PTH(IZF+1,3) = PTH(IZF,4)                                       CPDS 289
001376          GO TO 5025                                                      CPDS 290
001377       56 PTH(IZF,1) = PTH(IZF-1,1)                                       CPDS 291
001401     5025 NBH = NTOT                                                      CPDS 292
001403          KI = 1                                                          CPDS 293
001404          DO 40 I = 1, NTOT                                               CPDS 294
001405          IF (I .EQ. 1) GO TO 39                                          CPDS 295
001407          IF (ABS(PTH(I,1)-PTH(I-1,1)).GT..0001) GO TO 39                 CPDS 296
001414          PTH(KI-1,4) = PTH(I,4)                                          CPDS 297
001416          NBH = NBH - 1                                                   CPDS 298
001420          GO TO 40                                                        CPDS 299
001420       39 PTH(KI,1) = PTH(I,1)                                            CPDS 300
001423          PTH(KI,2) = PTH(I,2)                                            CPDS 301
001424          PTH(KI,3) = PTH(I,3)                                            CPDS 302
001426          PTH(KI,4) = PTH(I,4)                                            CPDS 303
001427          KI = KI + 1                                                     CPDS 304
001431       40 CONTINUE                                                        CPDS 305
001434          NC = NBH                                                        CPDS 306
001435          IF (IPTIO .LT. 3) GO TO 101                                     CPDS 307
001440          WRITE (6,1038)                                                  CPDS 308
001443     1038 FORMAT (1H1,10X,44HPTH ARRAY AFTER CUTOFF AND LENGTH ADJUSTMENT//)CPDS 309
001443          DO 1009 I = 1, NC                                               CPDS 310
001445     1009 WRITE (6,1002) (PTH(I,K), K = 1, 4)                             CPDS 311
001463          IF (KIPPER .EQ. 1) WRITE (6,5800) KPF,KIPF,KPS,KIPS             CPDS 312
001500     5800 FORMAT (1H0,10X,87HTHE FLOPT SUBROUTINE IS CALLED A NUMBER OF TIMECPDS 313
               1S BECAUSE THE END PLATES ARE TOO SHORT. /15X,6HKPF = ,I1,2X,    CPDS 314
               27HKIPF = ,I2,2X,6HKPS = ,I2,2X,7HKIPS = ,I2)                    CPDS 315
           C          CALL ON FLOPT TO DETERMINE THE SPLICE LOCATIONS           CPDS 316
001500      101 CALL FLOPT (PTH,NC,COST,FT)                                     CPDS 317
001503          IF (KIPPER .EQ. 0) GO TO 5100                                   CPDS 318
001504          IF (FT .GT. FTOT) GO TO 5110                                    CPDS 319
001510          DO 5111 I = 1, 40                                               CPDS 320
001511          DO 5111 K = 1, 4                                                CPDS 321
001512     5111 PSH(I,K) = 0.0                                                  CPDS 322
001522          DO 5112 I = 1, NC                                               CPDS 323
001523          DO 5112 K = 1, 4                                                CPDS 324
001524     5112 PSH(I,K) = PTH(I,K)                                             CPDS 325
001537          NTO = NC                                                        CPDS 326
001540          FTO = FT                                                        CPDS 327
001541          IF (IPTIO .EQ. 4) WRITE (6,81)                                  CPDS 328
001547       81 FORMAT (1H1,10X,24HTHE CURRENT PSH ARRAY...//)                  CPDS 329
001547          IF (IPTIO .NE. 4) GO TO 82                                      CPDS 330
001551          DO 83 I = 1, NTO                                                CPDS 331
```

```
001553          83 WRITE (6,1002) (PSH(I,K), K = 1, 4)                          CPDS 332
001571          82 IF (IPTIO .GE. 3) WRITE (6,5801) FTOT                        CPDS 333
001602        5801 FORMAT (1H0,30X,7HFTOT = ,F15.5)                             CPDS 334
001602        5110 IF (KIPF .EQ. 0) GO TO 5120                                  CPDS 335
001603             KPF = KPF - 1                                                CPDS 336
001605             IF (KPF .EQ. 0) GO TO 5120                                   CPDS 337
001606             GO TO 5300                                                   CPDS 338
001606        5120 IF (KIPS .EQ. 0) GO TO 5400                                  CPDS 339
001607             KPS = KPS - 1                                                CPDS 340
001611             IF (KPS .EQ. 0) GO TO 5400                                   CPDS 341
001612             KPF = KIPF                                                   CPDS 342
001613        5300 DO 5310 I = 1, 40                                            CPDS 343
001615             DO 5310 K = 1, 4                                             CPDS 344
001616        5310 PTH(I,K) = 0.0                                               CPDS 345
001626             DO 5320 I = 1, NTOT                                          CPDS 346
001627             DO 5320 K = 1, 4                                             CPDS 347
001630        5320 PTH(I,K) = PHH(I,K)                                          CPDS 348
001643             GO TO 5500                                                   CPDS 349
001644        5400 DO 5440 I = 1, 40                                            CPDS 350
001646             DO 5440 K = 1, 4                                             CPDS 351
001647        5440 PTH(I,K) = 0.0                                               CPDS 352
001657             NC = NTO                                                     CPDS 353
001660             DO 5441 I = 1, NC                                            CPDS 354
001662             DO 5441 K = 1, 4                                             CPDS 355
001663        5441 PTH(I,K) = PSH(I,K)                                          CPDS 356
001676        5500 KH = 0                                                       CPDS 357
            C          SET UP THE COVER PLATE ARRAY                             CPDS 358
001677             DO 8000 I = 1, NC                                            CPDS 359
001701             IF (PTH(I,1) .LT. .1) GO TO 8000                             CPDS 360
001704             IF (ABS(PTH(I,1)-PTH(I-1,1)) .LT. .00001) GO TO 8003         CPDS 361
001710             KH = KH + 1                                                  CPDS 362
001712             COVPL(KH,1,J) = PTH(I,3)                                     CPDS 363
001716             COVPL(KH,2,J) = PTH(I,4)                                     CPDS 364
001721             COVPL(KH,3,J) = PTH(I,1)                                     CPDS 365
001724             COVPL(KH,4,J) = PTH(I,2)                                     CPDS 366
001727             GO TO 8000                                                   CPDS 367
001730        8003 COVPL(KH,2,J) = PTH(I,4)                                     CPDS 368
001735        8000 CONTINUE                                                     CPDS 369
            C          PRINT THE COVPL ARRAY IF REQUIRED                        CPDS 370
001740             IF (IPTIO .LT. 3) GO TO 102                                  CPDS 371
001742             WRITE (6,1050)                                               CPDS 372
001746        1050 FORMAT (1H0,10X,16HCOVPL ARRAY.....//)                       CPDS 373
001746             DO 1051 I = 1, 12                                            CPDS 374
001750        1051 WRITE (6,1053) (COVPL(I,K,J), K = 1, 4)                      CPDS 375
001767        1053 FORMAT (10X,4(F15.5,5X))                                     CPDS 376
            C          SET UP THE PLATH AND PLWID ARRAYS                        CPDS 377
001767         102 DO 6110 I = 1, NF                                            CPDS 378
001771             IF (MN .EQ. 1) GO TO 10                                      CPDS 379
001773             DO 6250 II = 2, MN                                           CPDS 380
001774             ID = IDSP(II)                                                CPDS 381
001776             IF (I .EQ. ID-1 .OR. I .EQ. ID) GO TO 6110                   CPDS 382
002005        6250 CONTINUE                                                     CPDS 383
002007          10 DT = (COOR(I) + COOR(I+1)) / 2.                              CPDS 384
002013             DO 6109 JJ = 1, 12                                           CPDS 385
002014             IF (ABS(COVPL(JJ,1,J)) .LT. .00001 .AND. ABS(COVPL(JJ,2,J))  CPDS 386
                  1.LT..00001) GO TO 6007                                       CPDS 387
002034             IF (DT .GE. COVPL(JJ,1,J) .AND. DT .LE. COVPL(JJ,2,J)) GO TO 6108 CPDS 388
002046        6109 CONTINUE                                                     CPDS 389
```

```
002053     6007 PLATH(I,J) = 0.00                                                  CPDS 390
002054          PLWID(I,J) = 0.000                                                 CPDS 391
002056          GO TO 6110                                                         CPDS 392
002057     6108 CONTINUE                                                           CPDS 393
002057          IF (ABS(DT - COVPL(JJ,1,J)) .LT. .00001) GO TO 4001                CPDS 394
002066          IF (ABS(DT - COVPL(JJ,2,J)) .LT. .00001) GO TO 4000                CPDS 395
002075          PLATH(I,J) = COVPL(JJ,3,J)                                         CPDS 396
002103          PLWID(I,J) = COVPL(JJ,4,J)                                         CPDS 397
002110          GO TO 6110                                                         CPDS 398
002111     4000 KJST = JJ                                                          CPDS 399
002113          IF (COVPL(JJ+1,3,J) .LT. COVPL(JJ,3,J)) KJST = JJ + 1              CPDS 400
002123          PLATH(I,J) = COVPL(KJST,3,J)                                       CPDS 401
002131          PLWID(I,J) = COVPL(KJST,4,J)                                       CPDS 402
002136          GO TO 6110                                                         CPDS 403
002137     4001 KJST = JJ                                                          CPDS 404
002141          IF (COVPL(JJ-1,3,J) .LT. COVPL(JJ,3,J)) KJST = JJ - 1              CPDS 405
002151          PLATH(I,J) = COVPL(KJST,3,J)                                       CPDS 406
002157          PLWID(I,J) = COVPL(KJST,4,J)                                       CPDS 407
002164     6110 CONTINUE                                                           CPDS 408
002167          IF (J .EQ. 1) GO TO 6200                                           CPDS 409
002171          IF (JPT3 .GT. 9) GO TO 6350                                        CPDS 410
002174          J = 1                                                              CPDS 411
         C         SET UP THE TOP PLATES OVER THE INTERIOR SUPPORTS               CPDS 412
002174          DO 6140 I = 1, 10, 2                                               CPDS 413
002176          IF (LPCM(I) .EQ. 0) GO TO 6145                                     CPDS 414
002177          IA = LPCM(I) - 1                                                   CPDS 415
002201          IB = LPCM(I+1)                                                     CPDS 416
002203          DO 6141 JJ = IA, IB                                                CPDS 417
002205          PLATH(JJ,1) = PLATH(JJ,2)                                          CPDS 418
002207     6141 PLWID(JJ,1) = PLWID(JJ,2)                                          CPDS 419
002212     6140 CONTINUE                                                           CPDS 420
002214     6145 CALL SEMIC                                                         CPDS 421
         C         PRINT THE SECTION INFORMATION IF REQUIRED                      CPDS 422
002215          IF (IPT10 .LT. 4) GO TO 6300                                       CPDS 423
002220          CALL PRSEC (NF,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CL,IHCON,CH,   CPDS 424
               1COOH,SUHLEN,0,NCY)                                                 CPDS 425
002236          GO TO 6300                                                         CPDS 426
002237     6350 DO 6251 I = 1, 12                                                  CPDS 427
002241          DO 6251 K = 1, 6                                                   CPDS 428
002242     6251 COVPL(I,K,1) = COVPL(I,K,2)                                        CPDS 429
002255          DO 6252 I = 1, NF                                                  CPDS 430
002256          PLATH(I,1) = PLATH(I,2)                                            CPDS 431
002260     6252 PLWID(I,1) = PLWID(I,2)                                            CPDS 432
002263     6200 RETURN                                                             CPDS 433
002264          END                                                                CPDS 434

SUBPROGRAM LENGTH
003304

UNUSED COMPILER SPACE
032500


                SUBROUTINE CPWDES                                                  CPWD    1
```

```
             C           DESIGNS ALL OF THE COVER PLATE WELDS                              CPWD
000002              COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18),     CPWD
                   1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),           CPWD
                   2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,        CPWD
                   3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),FM,SPAN(4),IOSP(5),    CPWD
                   4TMOMI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLO(10),PROP(3),CPWD
                   5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),REAR(5,3),CPWD
                   6COST(9),CONSP(16,3,4),COVPL(12,6,2)                                  CPWD
000002              COMMON/ONE/JPT3,IPT10                                                 CPWD
000002              COMMON/SIX/FU                                                         CPWD
000002              COMMON/EIGHT/IST                                                      CPWD
000002              REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH,ILINT                         CPWD
000002              DIMENSION TMST(4), TS(4)                                              CPWD
             C           DETERMINE THE ALLOWABLE WELD STATIC STRESS                        CPWD
000002              FV = 12.4                                                             CPWD
000004              IF (ABS(PROP(3) - 36.) .GT. 1.45) FV = 14.5                           CPWD
             C           BEGIN FOR THE TOP PLATES                                          CPWD
000012              K = 1                                                                 CPWD
000013              IM = IST                                                              CPWD
000015              IF (JPT3 .LT. 9) SLARA = SLABWD * SLABTH                              CPWD
000021          930 DO 780 I = 1, 12                                                      CPWD
000023              IF (ABS(COVPL(I,1,K)).LT..0001.AND.ABS(COVPL(I,2,K)).LT..0001)        CPWD
                   1GO TO 780                                                             CPWD
             C           DETERMINING STARTING AND ENDING NODES                             CPWD
000043              IA = 0                                                                CPWD
000044              IB = 0                                                                CPWD
000045              DO 781 J = 1, NF                                                      CPWD
000046              IF (IA .NE. 0) GO TO 782                                              CPWD
000047              IF (COVPL(I,1,K).GE.COOR(J).AND.COVPL(I,1,K).LE.COOR(J+1)) IA=J+1 CPWD
000065          782 IF (COVPL(I,2,K).GE.COOR(J).AND.COVPL(I,2,K).LE.COOR(J+1)) IB=J   CPWD
000102              IF (IA .NE. 0 .AND. IB .NE. 0) GO TO 783                              CPWD
000110          781 CONTINUE                                                              CPWD
             C           DETERMINE THE END CONDITIONS OF THE PLATE                         CPWD
000112          783 IF (I .EQ. 1) GO TO 801                                               CPWD
000114              IF (I .EQ. 12) GO TO 802                                              CPWD
000116              IF (ABS(COVPL(I,1,K)-COVPL(I-1,2,K)).GT..0001) GO TO 801              CPWD
000127              IF (ABS(COVPL(I,2,K)-COVPL(I+1,1,K)).GT..0001) GO TO 802              CPWD
000136              COVPL(I,5,K) = 0.00                                                   CPWD
000140              GO TO 999                                                             CPWD
000141          801 IF (ABS(COVPL(I,2,K)-COVPL(I+1,1,K)).GT..0001) GO TO 830              CPWD
000153          837 D = COVPL(I,1,K) + 1.5*COVPL(I,4,K) - COOR(IA)                        CPWD
000164              KT = 1.0                                                              CPWD
000166              IF (D .LT. 0.0) KT = 2.0                                              CPWD
000171              IT = IA                                                               CPWD
000173              IS = IA                                                               CPWD
000174              DP = ABS(D)                                                           CPWD
000175              GO TO 7000                                                            CPWD
000176          802 IF (ABS(COVPL(I,1,K)-COVPL(I-1,2,K)).GT..0001) GO TO 830              CPWD
000210          835 D = COVPL(I,2,K) - 1.5*COVPL(I,4,K) - COOR(IB)                        CPWD
000221              KT = 1.0                                                              CPWD
000223              IF (D .LT. 0.0) KT = 2.0                                              CPWD
000226              IT = IB                                                               CPWD
000230              IS = IB - 1                                                           CPWD
000232              DP = ABS(D)                                                           CPWD
000233              GO TO 7000                                                            CPWD
000234          830 DO 831 J = 1, 4                                                       CPWD
000236          831 TMST(J) = DESMOM(IA,J)                                                CPWD
000245              IL = STRMAX(TMST)                                                     CPWD
```

```
000247            DO 832 J = 1, 4                                                            CPWD  60
000250        832 TMST(J) = DESMOM(IB, J)                                                    CPWD  61
000257            TR = STRMAX(TMST)                                                          CPWD  62
000261            IF (ABS(TR) .GT. ABS(TL)) GO TO 835                                        CPWD  63
000266            GO TO 837                                                                  CPWD  64
         C           DETERMINE THE WELD DESIGN STRESSES                                      CPWD  65
000266      7000 DO 702 JK = 1, 4 .                                                          CPWD  66
000270            DO 703 J = 1, NA                                                           CPWD  67
000271        703 RO(J) = DESMOM(J,JK)                                                       CPWD  68
000300        702 TMST(JK) = ILINT(IT,RO,COOR,KT,DP,LENGTH,SUB,FN)                           CPWD  69
000313            IF (K .EQ. 1) GO TO 710                                                    CPWD  70
000315            TMST(1) = TMST(1) * CS(IS) / ISTEEL(IS)                                    CPWD  71
000320            IF (JPT3 .GT. 9) GO TO 100                                                 CPWD  72
000324            TMST(2) = TMST(2) * CH(IS) / IHCON(IS)                                     CPWD  73
000326            TMST(3) = TMST(3) * CL(IS) / ILCON(IS)                                     CPWD  74
000331            TMST(4) = TMST(4) * CH(IS) / IHCON(IS)                                     CPWD  75
000333            GO TO 720                                                                  CPWD  76
000334        710 TD = DEPTH(IST) + PLATH(IS,1) + PLATH(IS,2)                                CPWD  77
000340            TMST(1) = TMST(1) * (TD - CS(IS)) / ISTEEL(IS)                             CPWD  78
000344            IF (JPT3 .GT. 9) GO TO 100                                                 CPWD  79
000347            TMST(2) = TMST(2) * (TD - CH(IS)) / IHCON(IS)                              CPWD  80
000352            TMST(3) = TMST(3) * (TD - CL(IS)) / ILCON(IS)                              CPWD  81
000354            TMST(4) = TMST(4) * (TD - CH(IS)) / IHCON(IS)                              CPWD  82
000361            GO TO 720                                                                  CPWD  83
000362        100 TMST(2) = TMST(2) *  CS(IS) / ISTEEL(IS)                                   CPWD  84
000365            TMST(3) = 0.0                                                              CPWD  85
000366            TMST(4) = TMST(4) * CS(IS) / ISTEEL(IS)                                    CPWD  86
000370        720 TDS = STRMAX(TMST)                                                         CPWD  87
000373            F = ABS(TDS * COVPL(I,4,K) * COVPL(I,3,K))                                 CPWD  88
         C           DETERMINE THE ALLOWABLE STRESSES                                        CPWD  89
000402            SL = TMST(1) + TMST(2) + TMST(3)                                           CPWD  90
000405            SR = TMST(1) + TMST(3) + TMST(4)                                           CPWD  91
000407            IF (ABS(SL) .GT. .0001 .AND. ABS(SR) .GT. .0001) GO TO 600                 CPWD  92
000423            R = 0.0                                                                    CPWD  93
000424            GO TO 610                                                                  CPWD  94
000424        600 R = SL / SR                                                                CPWD  95
000426            IF (ABS(SL) .GT. ABS(SR)) R = SR / SL                                      CPWD  96
000432        610 SK = 1.0 + .36 * (FJ/58. - 1.0)                                            CPWD  97
000437            AFS = 10.8 * SK / (1.0 - .55 * R)                                          CPWD  98
000443            ASTR = FV                                                                  CPWD  99
000445            IF (AFS .LT. ASTR) ASTR = AFS                                              CPWD 100
         C           DETERMINE THE SIZE OF THE WELD                                          CPWD 101
000450            TWS = F / (2.828 * COVPL(I,4,K) * ASTR)                                    CPWD 102
000456            N16 = TWS / .0625                                                          CPWD 103
000461            IF (ABS(TWS - N16*.0625) .GT. .0001) N16 = N16 + 1                         CPWD 104
000470            TWS = .0625 * N16                                                          CPWD 105
000472            TMAX = FLTH(IST)                                                           CPWD 106
000474            TTM = FLTH(IST)                                                            CPWD 107
000476            IF (COVPL(I,3,K) .GT. TMAX) TMAX = COVPL(I,3,K)                            CPWD 108
000506            IF (COVPL(I,3,K) .LT. TMAX) TTM = COVPL(I,3,K)                             CPWD 109
000516            TMIN = TOPT(TMAX)                                                          CPWD 110
         C           STORE THE REQUIRED WELD SIZE                                            CPWD 111
000521            IF (TWS .GT. TMIN) GO TO 500                                               CPWD 112
000524            IF (TTM .LT. TMIN) GO TO 503                                               CPWD 113
000526            COVPL(I,5,K) = TMIN                                                        CPWD 114
000531            GO TO 400                                                                  CPWD 115
000532        503 COVPL(I,5,K) = TMIN                                                        CPWD 116
000536            IF (TWS .LT. TTM) COVPL(I,5,K) = TTM                                       CPWD 117
```

```
000544          GO TO 900                                                        CPWD 118
000545      500 COVPL(I,5,K) = TWS                                               CPWD 119
          C          CALCULATE THE THICKNESS OF THE SEAL WELD                    CPWD 120
000551      900 CPA = COVPL(I,3,K) * COVPL(I,4,K)                                CPWD 121
000557          DSS = 0.0                                                        CPWD 122
          C          DETERMINE THE POINT OF MAXIMUM SHEAR                        CPWD 123
000560          DO 750 J = IA, IB                                                CPWD 124
000562          IL = 2 * J - 2                                                   CPWD 125
000564          IR = 2 * J - 1                                                   CPWD 126
000566          DO 751 JK = 1, 4                                                 CPWD 127
000567      751 TMST(JK) = DESSH(IL,JK)                                          CPWD 128
000576          TL = STRMAX(TMST)                                                CPWD 129
000600          DO 752 JK = 1, 4                                                 CPWD 130
000601      752 TMST(JK) = DESSH(IR,JK)                                          CPWD 131
000610          TR = STRMAX(TMST)                                                CPWD 132
000612          IF (ABS(TL) .LT. DSS) GO TO 753                                  CPWD 133
000615          DSS = ABS(TL)                                                    CPWD 134
000616          IC = J - 1                                                       CPWD 135
000620          IT = J                                                           CPWD 136
000621          ID = IL                                                          CPWD 137
000622      753 IF (ABS(TR) .LT. DSS) GO TO 750                                  CPWD 138
000626          DSS = ABS(TR)                                                    CPWD 139
000627          IC = J                                                           CPWD 140
000630          IT = J                                                           CPWD 141
000631          ID = IR                                                          CPWD 142
000632      750 CONTINUE                                                         CPWD 143
000635          IF (DESMOM(IT,1) .LT. 0.0) GO TO 3000                            CPWD 144
000637          IF (JPT3 .GT. 9) GO TO 3000                                      CPWD 145
000642          IF (K .EQ. 2) GO TO 3012                                         CPWD 146
          C          DETERMINE THE TOP DESIGN STRESSSES IN THE POSITIVE REGION   CPWD 147
000644          TS(1) = DESSH(ID,1)*CPA*(DEPTH(IM)+PLATH(IC,2)-CS(IC)+           CPWD 148
               1PLATH(IC,1)/2.) / ISTEEL(IC)                                     CPWD 149
000655          TS(2) = DESSH(ID,2)*SLABA*(DEPTH(IM)+PLATH(IC,2)-CH(IC)+SLABTH/2.+CPWD 150
               1HAUNCH) / (MR * IHCON(IC))                                       CPWD 151
000671          TS(3) = DESSH(ID,3)*SLABA*(DEPTH(IM)+PLATH(IC,2)-CL(IC)+SLABTH/2.+CPWD 152
               1HAUNCH) / (3.0 * MR * ILCON(IC))                                 CPWD 153
000705          TS(4) = DESSH(ID,4)*SLABA*(DEPTH(IM)+PLATH(IC,2)-CH(IC)+SLABTH/2.+CPWD 154
               1HAUNCH) / (MR * IHCON(IC))                                       CPWD 155
000721          GO TO 3001                                                       CPWD 156
          C          DETERMINE THE BOTTOM DESIGN STRESSES IN THE POSITIVE REGION CPWD 157
000722     3012 TS(1) = DESSH(ID,1)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CS(IC))/       CPWD 158
               1ISTEEL(IC)                                                       CPWD 159
000733          TS(2) = DESSH(ID,2)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CH(IC))/       CPWD 160
               1IHCON(IC)                                                        CPWD 161
000743          TS(3) = DESSH(ID,3)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CL(IC))/       CPWD 162
               1ILCON(IC)                                                        CPWD 163
000754          TS(4) = DESSH(ID,4)*CPA*(DEPTH(IM)+PLATH(IC,2)/2.-CH(IC))/       CPWD 164
               1IHCON(IC)                                                        CPWD 165
000764          GO TO 3001                                                       CPWD 166
          C          DETERMINE THE DESIGN STRESSES IN THE NEGATIVE REGION        CPWD 167
000765     3000 IF (K .EQ. 2) GO TO 3010                                         CPWD 168
000767          CONST = CPA * (DEPTH(IM) + PLATH(IC,2) - CS(IC) + PLATH(IC,1)/2.) CPWD 169
               1/ ISTEEL(IC)                                                     CPWD 170
000777          GO TO 3011                                                       CPWD 171
001000     3010 CONST = CPA * (DEPTH(IM) + PLATH(IC,2)/2. - CS(IC))/ISTEEL(IC)   CPWD 172
001007     3011 TS(1) = DESSH(ID,1) * CONST                                      CPWD 173
001012          TS(2) = DESSH(ID,2) * CONST                                      CPWD 174
001013          TS(3) = DESSH(ID,3) * CONST                                      CPWD 175
```

```
001015            TS(4) = DESSH(10,4) * CONST                                      CPWD 176
         C           DETERMINE THE TOTAL DESIGN STRESS                             CPWD 177
001016      3001  TSO = STFMAX(TS)                                                 CPWD 178
         C           CALCULATE THE FATIGUE ALLOWABLE STRESS                        CPWD 179
001021            SL = TS(1) + TS(2) + TS(3)                                       CPWD 180
001024            SR = TS(1) + TS(3) + TS(4)                                       CPWD 181
001026            IF (ABS(SL).GT..0001 .AND. ABS(SR).GT..0001) GO TO 2700          CPWD 182
001040            AFS = SR * 10.6                                                  CPWD 183
001042            GO TO 2701                                                       CPWD 184
001042      2700  R = SL / SR                                                      CPWD 185
001044            IF (ABS(SL) .GT. ABS(SR)) R = SR / SL                            CPWD 186
001050            AFS = SR * 10.6 / (1.0 - .55 * R)                                CPWD 187
001055      2701  ASST = FV                                                        CPWD 188
001057            IF (AFS .LT. ASST) ASST = AFS                                    CPWD 189
         C           DETERMINE THE WELD THICKNESS                                  CPWD 190
001062            TWS = ABS(TSO) / (1.414 * ASST)                                  CPWD 191
001067            N16 = TWS / .0625 + 1                                            CPWD 192
001073            IF (ABS(TWS - (N16 - 1) * .0625) .LT. .0001) N16 = N16 - 1       CPWD 193
001103            TWS = N16 * .0625                                                CPWD 194
001105            COVPL(I,6,K) = TWS                                               CPWD 195
001111            IF (TWS .LT. .1875) COVPL(I,6,K) = .1875                         CPWD 196
001116       780  CONTINUE                                                         CPWD 197
001120       790  IF (K .EQ. 2) GO TO 840                                          CPWD 198
001122            IF (JPT3 .GT. 8) GO TO 101                                       CPWD 199
001126            K = K + 1                                                        CPWD 200
001127            GO TO 930                                                        CPWD 201
001127       101  DO 102 II = 1, 12                                                CPWD 202
001131            COVPL(II,5,2) = COVPL(II,5,1)                                    CPWD 203
001133       102  COVPL(II,6,2) = COVPL(II,6,1)                                    CPWD 204
001136       840  RETURN                                                           CPWD 205
001137            END                                                              CPWD 206

SUBPROGRAM LENGTH
001330

UNUSED COMPILER SPACE
036500


                  SUBROUTINE DEFLEC                                                DEFL   1
         C           CALCULATE THE REQUIRED DEFLECTIONS FOR THE FINAL DESIGN       DEFL   2
000002            COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), DEFL   3
                 1DAF(18),IX(18),IY(18),NSECT(18),NDS(80),ISTEEL(80),CS(80),       DEFL   4
                 2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,    DEFL   5
                 3SLABWD,SLABTH,MR,COOR(81),BIL(5,8,3),CC(5,5),FM,SPAN(4),IOSP(5), DEFL   6
                 4TMORI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLD(9),TRLD(10),PROP(3),DEFL   7
                 5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),REAR(5,3),DEFL   8
                 6CONST(9),CONSP(16,3,4),COVPL(12,6,2)                             DEFL   9
000002            COMMON/ONE/JPT3,IPT10                                            DEFL  10
000002            COMMON/D/IDEF(20),DESDEF(20,4),NDEF,PERCEN,NCYC                  DEFL  11
000002            COMMON/E/IPT2                                                    DEFL  12
000002            REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                          DEFL  13
000002            DIMENSION TIL(81,3), STORE(10)                                   DEFL  14
         C           ZERO OUT THE REQUIRED ARRAYS                                  DEFL  15
```

```
000002          DO 9800 I = 1, NA
000004          DO 9800 J = 1, 3
000005    9800 TIL(I,J) = 0.0
000015          DO 9801 I = 1, NDEF
000017          DO 9801 J = 1, 4
000020    9801 DESDEF(I,J) = 0.0
          C          PRINT THE HEADING IF REQUIRED
000030          IF (IPT10 .GE. 3) WRITE (6,9700)
000037    9700 FORMAT (1H1,22HDEFLECTION LOADINGS...//)
          C          START THE LOOP FOR EACH DEFLECTION REQUIRED
000037          DO 9500 I = 1, NDEF
000041          IA = IDEF(I)
000043          DO 9600 K = 1, 3
          C          DETERMINE THE BENDING MOMENTS DUE TO A UNIT LOAD
000044          IF (JPT3 .GT. 9 .AND. K .EQ. 2) GO TO 750
000054          DO 9510 J = 1, NA
000055          WR = 0.0
000056          DO 9511 JJ = 1, NS
000060          ID = IOSP(JJ)
000062          IF (J .LE. ID) GO TO 9511
000064          WR = WR + RIL(J,IA,K) * (COOR(J) - COOR(ID))
000075    9511 CONTINUE
000100          SP = 0.0
000101          IF (J .GT. IA) SP = COOR(J) - COOR(IA)
000105          RO(J) = WR - SP
000110    9510 CONTINUE
          C          DETERMINE THE M/EI ARRAY
000112          DO 9515 J = 1, NF
000114          UI = ISTFEL(J)
000116          IF (K .EQ. 2) UI = ILCON(J)
000121          IF (K .EQ. 3) UI = IHCON(J)
000125          IMOEI(J) = (RO(J) + RO(J+1)) / (2. * EM * UI)
000133    9515 CONTINUE
          C          DETERMINE THE TANGENT AT THE LEFT END
000136          WP = 0.0
000137          DO 9516 J = 1, NF
000140          WP = WP + IMOEI(J) * SUBLEN(J) * (LENGTH - COOR(J) - SUBLEN(J)/2.)
000146    9516 CONTINUE
000150          TALE = WP / LENGTH
          C          DETERMINE THE DEFLECTION AT EACH ANALYSIS POINT
000152          TIL(1,K) = 0.0
000155          TIL(NA,K) = 0.0
000160          DO 9520 J = 2, NF
000162          WP = 0.0
000163          IB = J - 1
000165          DO 9521 JJ = 1, IB
          C          STORE THE DEFLECTIONS AS THE INFLUENCE COEFFICIENTS
000166    9521 WP = WP+IMOEI(JJ)*SUBLEN(JJ)*(COOR(J)-COOR(JJ)-SUBLEN(JJ)/2.)
000177    9520 TIL(J,K) = TALE * COOR(J) - WP
000207    9600 CONTINUE
000211     750 CONTINUE
          C          LOAD THE INFLUENCE LINES TO DETERMINE THE DEFLECTIONS
000211          CALL LOAD(TIL,TRLO,JNLO,SUBLEN,NA,IPT2,LENGTH,COOR,IA,4HDEFL,NS,
               1IOSP,SPAN,STORE)
000227          IF (IPT10 .GE. 3) WRITE(6,9701) IA, (STORE(KK), KK = 1, 10)
000246    9701 FORMAT (I5,X,10(F11.8,X))
          C          DETERMINE THE DESIGN DEFLECTIONS
000246          DESDEF(I,1) = STORE(1)
```

```
000250          DESDEF(I,3) = STORE(10)                                   DEFL  74
000252          POS = 0.0                                                 DEFL  75
000253          SEG = 0.0                                                 DEFL  76
000254          DO 9550 K = 2, 7                                          DEFL  77
000255          IF (STORE(K) .GT. POS) POS = STORE(K)                     DEFL  78
000261          IF (STORE(K) .LT. SEG) SEG = STORE(K)                     DEFL  79
000265     9550 CONTINUE                                                  DEFL  80
000267          DESDEF(I,2) = POS + STORE(8)                              DEFL  81
000272          DESDEF(I,4) = SEG + STORE(9)                              DEFL  82
000275     9500 CONTINUE                                                  DEFL  83
          C        PRINT THE DESIGN DEFLECTIONS IF REQUIRED               DEFL  84
000277          IF (IPTIO .LT. 3) GO TO 20                                DEFL  85
000301          WRITE (6,9710)                                            DEFL  86
000305     9710 FORMAT (1H0,21HDESIGN DEFLECTIONS...//)                   DEFL  87
000305          DO 9711 I = 1, NDEF                                       DEFL  88
000307          IA = IDEF(I)                                              DEFL  89
000311     9711 WRITE (6,9702) IA, (DESDEF(I,K), K = 1, 4)                DEFL  90
000331     9702 FORMAT (I5,4X,4(F12.9,4X))                                DEFL  91
000331       20 CONTINUE                                                  DEFL  92
000331          RETURN                                                    DEFL  93
000332          END                                                       DEFL  94

SUBPROGRAM LENGTH
001013

UNUSED COMPILER SPACE
041100


                SUBROUTINE EQSET(RO,SUBLEN,LENGTH,COOR,NF,TMI,EM,NS,IOSP,SPAN,RI) EQST  1
          C        SOLVES FOR THE INITIAL VALUES IN THE RII MATRIX WHICH ARE THE  EQST  2
          C        DEFLECTIONS OF A SIMPLE BEAM AT THE SUPPORT POINTS         EQST  3
000016          REAL LENGTH                                               EQST  4
000016          DIMENSION RO(81),SUBLEN(80),COOR(80),TMI(80),IOSP(5),SPAN(4), EQST  5
               1RI(4),WA(5),WB(5),TMOEI(80)                               EQST  6
000016          DO 80 I = 1, 4                                            EQST  7
000017          WA(I) = 0.0                                               EQST  8
000020       80 WB(I) = 0.0                                               EQST  9
          C        CALCULATE TANGENT AT THE LEFT END                      EQST 10
000023          WP = 0.0                                                  EQST 11
000024          DO 81 J = 1, NF                                           EQST 12
000025          TMOEI(J) = (RO(J)+RO(J+1))/(2.*TMI(J)*EM)                 EQST 13
000036       81 WP = WP+TMOEI(J)*SUBLEN(J)*(LENGTH-COOR(J)-SUBLEN(J)/2.)  EQST 14
000050          TANL = WP/LENGTH                                          EQST 15
          C        CALCULATE DEFLECTIONS AT THE INTERIOR SUPPORTS         EQST 16
000051          MN = NS - 1                                               EQST 17
000053          DO 82 J = 2, MN                                           EQST 18
000055          IA = IOSP(J-1)                                            EQST 19
000057          IB = IOSP(J) - 1                                          EQST 20
000062          IC = IOSP(J)                                              EQST 21
000065          DO 83 K = IA, IB                                          EQST 22
000067          WA(J) = WA(J) + TMOEI(K) * SUBLEN(K)                      EQST 23
000074       83 WB(J) = WB(J)+TMOEI(K)*SUBLEN(K)*(COOR(IC)-COOR(K)-SUBLEN(K)/2.) EQST 24
000112          WA(J) = WA(J) + WA(J-1)                                   EQST 25
```

```
000114             WH(J) = WH(J) + WH(J-1) + WA(J-1)*SPAN(J-1)                                   EQST  26
000122          92 PI(J-1) = COOR(IC) * TALE - WH(J)                                             EQST  27
000133             RETURN                                                                        EQST  28
000133             END                                                                           EQST  29

SUBPROGRAM LENGTH
000334

UNUSED COMPILER SPACE
042400



                   SUBROUTINE FLOPT (PTH,NC,COST,FT)                                             FOPT   1
             C        OPTIMIZE THE LOCATIONS OF THE COVER PLATE SPLICES                          FOPT   2
000007             COMMON/ONE/JPT3,IPT10                                                         FOPT   3
000007             DIMENSION STTH(40),PTH(40,4),PCTS(40),COST(9),CT(40),TP(80,40)                FOPT   4
000007             IF (IPT10 .GE. 3) WRITE(6,100)                                                FOPT   5
000020         100 FORMAT (1H1,10X,14HENTRY TO FLOPT//)                                          FOPT   6
000020         990 DO 1900 I = 1, NC                                                             FOPT   7
000022             IF (PTH(I,2) .LT. 4.0) GO TO 1900                                             FOPT   8
000025             PLW = PTH(I,2)                                                                FOPT   9
000027             GO TO 1902                                                                    FOPT  10
000027        1900 CONTINUE                                                                      FOPT  11
000032        1902 DO 970 I = 1, NC                                                              FOPT  12
000034         970 STTH(I) = 0.0                                                                 FOPT  13
             C        DETERMINE THE STTH ARRAY                                                   FOPT  14
000037             STTH(1) = PTH(1,1)                                                            FOPT  15
000040             K = 2                                                                         FOPT  16
000041             IB = 1                                                                        FOPT  17
000042             DO 971 I = 2, NC                                                              FOPT  18
000044             DO 972 KK = 1, IB                                                             FOPT  19
000045             IF (ABS(STTH(KK) - PTH(I,1))   .LT. .0001) GO TO 971                          FOPT  20
000052         972 CONTINUE                                                                      FOPT  21
000055             STTH(K) = PTH(I,1)                                                            FOPT  22
000057             K = K + 1                                                                     FOPT  23
000060             IB = K - 1                                                                    FOPT  24
000061         971 CONTINUE                                                                      FOPT  25
             C        PLACE THE STTH ARRAY IN ASCENDING ORDER                                    FOPT  26
000064             CALL SORT (STTH,IB)                                                           FOPT  27
000066             IF (IPT10 .LT. 3) GO TO 400                                                   FOPT  28
000073             WRITE (6,980)                                                                 FOPT  29
000077         980 FORMAT (1H0,10X,15HSTTH ARRAY.....//)                                         FOPT  30
000077             DO 981 I = 1, IB                                                              FOPT  31
000103         981 WRITE (6,982) I, STTH(I)                                                      FOPT  32
000117         982 FORMAT (10X,I2,5X,F10.4)                                                      FOPT  33
000117         400 DO 2900 I = 1, NC                                                             FOPT  34
000121             DO 2900 JK = 1, IB                                                            FOPT  35
000122        2900 TP(I,JK) = 0.0                                                                FOPT  36
000133             DO 2901 I = 1, IB                                                             FOPT  37
             C        DETERMINE THE STARTING SECTION                                             FOPT  39
000134        2901 PCTS(I) = 0.0                                                                 FOPT  38
000137             I = 1                                                                         FOPT  40
000140        2904 IF (ABS(PTH(I,1))   .GT. .0001) GO TO 2903                                    FOPT  41
000146             I = I + 1                                                                     FOPT  42
```

```
000147          GO TO 2904
000147    2903 IS = 1
          C        DETERMINE THE ENDING SECTION
000151          I = NC
000152    2906 IF (ABS(PTH(I,1))    .GT. .0001) GO TO 2905
000160          I = I - 1
000161          GO TO 2906
000161    2905 IF = I
          C        BEGIN THE OPTIMIZATION OF THE COVER PLATES
000163          DO 4000 I = IS, IF
000165          PLLEN = PTH(I,4) - PTH(I,3)
000170          IF (IPT(9) .EQ. 4) WRITE (6,101) IS,IE,I,PLW,PLLEN
000213     101 FORMAT(1H0,5HIS = ,I2,X,5HIE = ,I2,X,4HI = ,I2,X,6HPLW = ,F5.1,
              1X,8HPLLEN = ,F10.2)
          C        INITIALIZE THE STARTING CONSTANTS AND COSTS
000213          DO 4001 K = 1, IH
000215    4001 CT(K) = 0.0
000220          IF (I .EQ . IF) GO TO 4500
          C        VARY THE THICKNESS OF THE ADJACENT SECTION
000222          DO 4010 K = 1, IH
000224          TCT = 99999.
000226          IF (STTH(K) .GT. PTH(I+1,1))   GO TO 4020
000232          IF (ABS(STTH(K) - PTH(I+1,1))    .LT. .00001) GO TO 4020
000236          CT(K) = 0.0
000237          GO TO 4010
          C        VARY THE THICKNESS IN THE DESIGN SECTION
000240    4020 DO 4030 KK = 1, IH
000242          IF (STTH(KK) .GT. PTH(I,1))   GO TO 4040
000247          IF (ABS(STTH(KK) - PTH(I,1))    .LT. .00001) GO TO 4040
000253          GO TO 4030
000254    4040 IF (I .NE. IS) GO TO 4050
          C        DETERMINE THE COST OF THE INITIAL SECTION
000256          TMAX = TOPT (STTH(KK))
000262          CFW = COST(6)+PLW*COST(8)+.5*TMAX**2*PLW*COST(7)
000273          CSW = 2.*COST(6)+2.*  PLLEN  *(COST(8)+.017594*COST(7))
000301          IF (ABS(STTH(K)) .GT. .000001) GO TO 4041
000306          CBW = COST(6)+PLW*COST(8)+PLW*TMAX**2*COST(7)/2.
000315          GO TO 4042
          C        DETERMINE THE COST OF THE BUTT WELD IF REQUIRED
000315    4041 TH = STTH(K)
000317          IF (STTH(KK) .LT. STTH(K)) TH = STTH(KK)
000324          IF (ABS(STTH(KK) - STTH(K)) .LT. .00001) GO TO 4043
000331          CBW = COST(6)+PLW*(COST(8)+(.125*TH+.288675*TH**2)*COST(7))
000341          GO TO 4042
000342    4043 CBW = 0.0
          C        DETERMINE THE MATERIAL COST OF THE PLATE
000343    4042 CPT =   PLLEN  *PLW*STTH(KK)*COST(1)*490./1728.
          C        DETERMINE THE TOTAL COST OF THE SECTION
000350          TC = CPT + CBW + CFW + CSW + PCTS(KK)
000356          GO TO 4070
          C        DETERMINE THE COST OF THE SEAL WELDS
000356    4050 IF (ABS(STTH(KK)) .LT. .000001) GO TO 4055
000362          IF (ABS(STTH(K)) .LT. .00001) GO TO 4056
000366          CSW = 2.*  PLLEN  *(COST(8)+.017594*COST(7))
000373          CFW = 0.0
```

```
000374          GO TO 4041                                                    FOPT  98
000374     4055 IF (ABS(STTH(K)) .LT. .00001) GO TO 4057                       FOPT  99
          C         DETERMINE THE COST OF THE APPROXIMATE END WELDS            FOPT 100
000400          TMAX = TOPT(STTH(K))                                           FOPT 101
000403          CEW = COST(6)*3.*PLW*COST(8)+PLW*.5*TMAX**2*COST(7)            FOPT 102
000416          TC = CEW + PCTS(KK)                                            FOPT 103
000420          GO TO 4070                                                     FOPT 104
000421     4056 TMAX = TOPT(STTH(KK))                                          FOPT 105
000425          CEW = COST(6)+PLW*COST(8)+.5*TMAX**2*COST(7)*PLW               FOPT 106
000436          CSW = 0.0                                                      FOPT 107
000437          CSW = 2.*   PLLEN   *(COST(8)+.017594*COST(7))                 FOPT 108
000443          GO TO 4042                                                     FOPT 109
000444     4057 TC = PCTS(KK)                                                  FOPT 110
000446     4070 IF (IPT10 .EQ. 4) WRITE (6,102) STTH(K), STTH(KK), TC          FOPT 111
000465      102 FORMAT (5X,21HTHICKNESS ON RIGHT = ,F10.5,2X,20HTHICKNESS ON LEFT FOPT 112
               1= ,F10.5,2X,7HCOST = ,F10.2)                                   FOPT 113
000465          IF (TC .GT. TC1) GO TO 4030                                    FOPT 114
          C         STORE THE COST AND THE THICKNESS                           FOPT 115
000471          TC1 = TC                                                       FOPT 116
000471          IKP = KK                                                       FOPT 117
000473     4030 CONTINUE                                                       FOPT 118
          C         STORE THE THICKNESS REQUIRED IN THE TP ARRAY               FOPT 119
000476          TP(I,K) = STTH(IKP)                                            FOPT 120
000502          CT(K) = TC1                                                    FOPT 121
000504          IF (IPT10 .EQ. 4) WRITE (6,110) CT(K), TP(I,K)                 FOPT 122
000522      110 FORMAT (1H0,10X,7HCOST = ,F15.5,5X,5HTP = ,F15.5//)            FOPT 123
000522     4010 CONTINUE                                                       FOPT 124
          C         DETERMINE THE PCTS ARRAY                                   FOPT 125
000525          DO 4090 KKK = 1, IH                                            FOPT 126
000526     4090 PCTS(KKK) = CT(KKK)                                            FOPT 127
000532          IF (IPT10 .LT. 4) GO TO 4000                                   FOPT 128
000534          WRITE (6,410)                                                  FOPT 129
000540      410 FORMAT (1H0,10HPCTS ARRAY//)                                   FOPT 130
000540          WRITE (6,103) (PCTS(KKK), KKK = 1, IH)                         FOPT 131
000553      103 FORMAT (4(F15.5,5X))                                           FOPT 132
000553     4000 CONTINUE                                                       FOPT 133
          C         VARY THE THICKNESS OF THE LAST SECTION                     FOPT 134
000560     4500 DO 4100 K = 1, IH                                              FOPT 135
000562          IF (STTH(K) .GT. PTH(IE,1))   GO TO 4120                       FOPT 136
000567          IF (ABS(STTH(K) - PTH(IE,1))   .LT. .000001) GO TO 4120        FOPT 137
000573          PCTS(K) = 0.0                                                  FOPT 138
000575          GO TO 4100                                                     FOPT 139
          C         DETERMINE THE COST OF THE LAST SECTION                     FOPT 140
000575     4120 CPC = STTH(K)*   PLW     * PLLEN  *   COST(1)*490./1728.       FOPT 141
000602          CSW = 2.*  PLLEN   *(COST(8)+.017594*COST(7))                  FOPT 142
000607          IF (IE .EQ. 15) CSW = CSW + 2. * COST(6)                       FOPT 143
000614          TMAX = TOPT(STTH(K))                                           FOPT 144
000620          CEW = COST(6)+    PLW    *COST(8)*.5*TMAX**2*COST(7)*  PLW     FOPT 145
000631          PCTS(K) = PCTS(K) + CPC + CSW + CEW                            FOPT 146
000636          IF (IE .EQ. 15) PCTS(K) = PCTS(K) + CEW                        FOPT 147
000642     4100 CONTINUE                                                       FOPT 148
000645          TCT = 99999.                                                   FOPT 149
          C         DETERMINE THE THICKNESS OF THE LAST SECTION                FOPT 150
000646          DO 4130 K = 1, IH                                              FOPT 151
000650          IF (ABS(PCTS(K)) .LT. .000001) GO TO 4130                      FOPT 152
000654          IF (PCTS(K) .GT. TC1) GO TO 4130                               FOPT 153
000657          TCT = PCTS(K)                                                  FOPT 154
000660          TCH = K                                                        FOPT 155
```

```
000660     4130 CONTINUE                                                         FOPT 156
000663          IF (IPT10 .LT. 3) GO TO 420                                      FOPT 157
000665          IT = IE - 1                                                      FOPT 158
000667          WRITE (6,450)                                                    FOPT 159
000673      450 FORMAT (1H0,10X,18HFINAL TP ARRAY....//)                         FOPT 160
000673          DO 5520 I = IS, IT                                               FOPT 161
000700     5520 WRITE (6,5521) I, ( TP(I,K), K = 1, 18)                          FOPT 162
000723     5521 FORMAT (5X,I2,5X,10(F7.4,3X))                                    FOPT 163
000723          WRITE (6,455)                                                    FOPT 164
000727      455 FORMAT (1H0,10X,38HCALCULATIONS FOR THE LAST SEGMENT.....//)     FOPT 165
000727          DO 5522 K = 1, IR                                                FOPT 166
000733     5522 WRITE (6,5523) K, STTH(K), PCTS(K)                               FOPT 167
000751     5523 FORMAT (5X,4HK = ,I2,2X,7HSTTH = ,F10.5,2X,7HCOST = ,F15.5)      FOPT 168
000751      420 CONTINUE                                                         FOPT 169
000751          PTH(IE,1) = STTH(ICR)                                            FOPT 170
000754          FT = PCTS(ICR)                                                   FOPT 171
000755          IF (IPT10 .GE. 3) WRITE (6,460) IE, STTH(ICR), FT                FOPT 172
001000      460 FORMAT (1H0,21HTHICKNESS IN SEGMENT ,I2,3H = ,F10.5,15H WITH A COSFOPT 173
              1T = ,F15.5)                                                       FOPT 174
        C          DETERMINE THE THICKNESSES OF THE REMAINING SECTIONS           FOPT 175
001000          I = IE                                                           FOPT 176
001002          IF (IS .EQ. IE) GO TO 4160                                       FOPT 177
001003     4170 DO 4140 K = 1, IR                                                FOPT 178
001005          ICR = K                                                          FOPT 179
001006          IF (ABS(STTH(K) - PTH(I,1))    .LT. .00001) GO TO 4150           FOPT 180
001013     4140 CONTINUE                                                         FOPT 181
001016          WRITE (6,5000)                                                   FOPT 182
001021     5000 FORMAT (1H0,10X,20HERROR FOUND IN FLOPT)                         FOPT 183
001021          STOP                                                             FOPT 184
001023     4150 PTH(I-1,1) = TP(I-1,ICR)                                         FOPT 185
001032          PTH(I-1,2) = PLW                                                 FOPT 186
001034          IF (ABS(PTH(I-1,1)) .LT. .00001) PTH(I-1,2) = 0.0                FOPT 187
001042          IF (I .EQ. IS + 1) GO TO 4160                                    FOPT 188
001045          I = I - 1                                                        FOPT 189
001046          GO TO 4170                                                       FOPT 190
001046     4160 CONTINUE                                                         FOPT 191
001046          IF (IPT10 .LT. 3) GO TO 470                                      FOPT 192
001051          WRITE (6,1100)                                                   FOPT 193
001054     1100 FORMAT (1H1,10X,18HPTH ARRAY IN FLOPT//)                         FOPT 194
001054          DO 1101 I = 1, NC                                                FOPT 195
001060     1101 WRITE (6,1102) PTH(I,1), PTH(I,2), PTH(I,3), PTH(I,4)            FOPT 196
001114     1102 FORMAT (2X,9HPLATE IS ,F10.5,3H X ,F6.2,10X,8HSTART = ,F10.2,10X, FOPT 197
              16HEND = ,F10.2)                                                   FOPT 198
001114          WRITE (6,104)                                                    FOPT 199
001117      104 FORMAT (1H0,10X,15HRETURN TO CPDES//)                            FOPT 200
001117      470 CONTINUE                                                         FOPT 201
001117          RETURN                                                           FOPT 202
001120          END                                                              FOPT 203


SUBPROGRAM LENGTH
007770

UNUSED COMPILER SPACE
037000
```

```
                   SUBROUTINE ICALC (A,B,D,C,H,AS,IS,HP,HA,LEF,ITOT)
              C       CALCULATES MOMENT OF INERTIA FOR A PARTICULAR SECTION
000016             REAL ITOT, LEF, IPART, IS
              C       CHECKS FOR COVER PLATES
              C       CALCULATES I FOR NO COVER PLATES
000016             IF (A .GT. .0001 .OR. D .GT. .0001) GO TO 10
000027             ITOT = IS
000030             LEF = HP
000032             RETURN
              C       SETS INITIAL ZERO VALUES
000033          10 ALP = 0.0
000034             AUP = 0.0
000034             DL = 0.0
000035             DU = 0.0
              C       CALCULATES VALUES FOR THE UPPER PLATE OR SLAB
000036             IF (A .LT. .0001) GO TO 11
000041             AUP = A * B
000042             DU = HA + (H + B)/2.
              C       CALCULATE VALUES FOR THE LOWER PLATE
000046          11 IF (D .LT. .0001) GO TO 12
000051             ALP = D * C
000052             DL = (H + C)/2.
000055          12 ATOT = AS + AUP + ALP
              C       CALCULATE DISTANCE FROM CENTROID OF STEEL SECTION TO THE
              C       CENTROID OF THE OVERALL SECTION, ALSO CALCULATE I FOR SECTION.
000060             DIST = (AUP * DU - ALP * DL)/ATOT
000064             LEF = HP + DIST
000066             IPART = IS + AUP*B*B/12. + ALP*C*C/12.
000076             ITOT = IPART + AUP*DU**2 + ALP*DL**2 - ATOT*DIST**2
000111             RETURN
000112             END

SUBPROGRAM LENGTH
000142

UNUSED COMPILER SPACE
042500


                   REAL FUNCTION ILINT (PT,RO,COOR,ISW,DP,LENGTH,SUBLEN)
              C       INTERPOLATES BETWEEN POINTS ON THE INFLUENCE LINE
000012             DIMENSION RO(81), COOR(81), SUBLEN(80)
000012             REAL LENGTH
000012             INTEGER PT
000012             IF (ISW .EQ. 2) GO TO 520
              C       CALCULATES VALUE OF INFLUENCE LINE TO THE RIGHT OF POINT PT
000014             SP = COOR(PT) + DP
              C       RETURNS A ZERO IF THE POINT IS OFF THE GIRDER
```

```
000016            IF (SP .GT. LENGTH) GO TO 510
000022            GO TO 512
000022        510 ILINT = 0.0
000023            RETURN
          C           DETERMINES THE SUBELEMENT LOCATION
000024        512 J = PT + 1
000026        513 IF (COOR(J) .GE. SP) GO TO 515
000032            J = J + 1
000033            GO TO 513
000034        515 K = J - 1
          C           INTERPOLATION FORMULA
000036            ILINT = RO(K) + (RO(J)-RO(K)) * (SP-COOR(K))/SUBLEN(K)
000052            RETURN
          C           CALCULATES VALUE OF INFLUENCE LINE TO THE LEFT OF POINT PT
000052        520 SP = COOR(PT) - OP
          C           RETURNS A ZERO IF THE POINT IS OFF THE GIRDER
000054            IF (SP .LT. 0.0) GO TO 522
000056            GO TO 525
000057        522 ILINT = 0.0
000060            RETURN
          C           DETERMINES THE SUBELEMENT LOCATION
000061        525 J = PT - 1
000063        527 IF (COOR(J) .LE. SP) GO TO 526
000067            J = J - 1
000070            GO TO 527
000071        526 K = J + 1
000073            J = K
000074            K = J - 1
          C           INTERPOLATION FORMULA
000075            ILINT = RO(K) + (RO(J)-RO(K)) * (SP-COOR(K))/SUBLEN(K)
000111            RETURN
000112            END

SUBPROGRAM LENGTH
000133

UNUSED COMPILER SPACE
042600


                  SUBROUTINE ILPROP (RO,SUBLEN,NA,KOP,ANEG,APOS,MAXOL,MINOL)
000013            DIMENSION RO(81), SUBLEN(80), KOP(10)
          C            DETERMINES THE PROPERTIES OF A GIVEN INFLUENCE LINE
000013            MAXOL = 1
000013            MINOL = 1
000015            ANEG = 1.0
000016            APOS = 0.0
000016            OMAX = -.0001
000017            OMIN = .0001
000021            DO 570 I = 1, 10
000023        570 KOP(I) = 0
          C           DETERMINE THE CROSSOVER POINTS
000026            KOP(1) = 1
000027            J = 2
```

```
000030          NE = NA - 1
000032          DO 550 I = 2, NE
000034          IF (ABS(RO(I)) .GT. 10.0**(-6)) GO TO 551
000043          KOP(J) = I
000045          J = J + 1
000047          GO TO 550
000047      551 IF (I .EQ. 1) GO TO 550
000051          IF (RO(I)*RO(I-1) .GT. 0.0) GO TO 550
000056          IF (ABS(RO(I-1)) .LT. 10.0**(-6)) GO TO 550
000063          KOP(J) = I
000065          J = J + 1
000067      550 CONTINUE
000072          KOP(J) = NA
000074          NKOP = J
           C        DETERMINE MAXIMUM AND MINIMUM ORDINATE LOCATIONS
000075          DO 552 I = 1, NA
000076          IF (RO(I) .GT. OMAX) GO TO 555
000102          IF (RO(I) .LT. OMIN) GO TO 554
000105          GO TO 552
000105      555 OMAX = RO(I)
000107          MAXOL = I
000111          GO TO 552
000112      554 OMIN = RO(I)
000114          MINOL = I
000116      552 CONTINUE
           C        DETERMINE THE POSITIVE AND NEGATIVE AREAS UNDER THE LINE
000121          DO 560 K = 1, NKOP
000122          IF (K .EQ. NKOP) GO TO 561
000124          IA = KOP(K)
000125          IB = KOP(K+1) - 1
000127          A = 0.0
000131          DO 562 I = IA, IB
000133          IF (I .NE. IA) GO TO 563
000135          IF (I .EQ. 1) GO TO 563
000137          IF (ABS(RO(I)) .LT. 10.0**(-6)) GO TO 563
000145          A = SUBLEN(I-1)*RO(I)*ABS(RO(I))/(2.*(ABS(RO(I))+ABS(RO(I-1))))
000160          GO TO 564
000161      563 IF (I .NE. IB) GO TO 564
000163          IF (I+1 .EQ. NA) GO TO 564
000165          IF (ABS(RO(I+1)) .LT. 10.0**(-6)) GO TO 564
000173          A = A+SUBLEN(I)*RO(I)*ABS(RO(I))/(2.*(ABS(RO(I))+ABS(RO(I+1))))
000210          GO TO 562
000211      564 A = A + (RO(I) + RO(I+1)) * SUBLEN(I)/2.0
000220      562 CONTINUE
000223          IF (A .GT. 0.0) APOS = APOS + A
000225      560 IF (A .LT. 0.0) ANEG = ANEG + A
000233      561 CONTINUE
000233          RETURN
000234          END

SUBPROGRAM LENGTH
000277


UNUSED COMPILER SPACE
042200
```

```
                  REAL FUNCTION IMPACT (TYPE,NP,IOSP,NS,SPAN,DEAD,COOR)
            C        DETERMINES THE IMPACT FACTOR
000012            DIMENSION IOSP(5), SPAN(4), COOR(81)
000012            INTEGER TYPE
000012            MN = NS - 1
000013            IF (MN .EQ. 1) GO TO 7000
000015            IF (TYPE .NE. 4HREAC) GO TO 600
            C        IMPACT DISTANCE FOR THE REACTION INFLUENCE LINES
000017            IF (NP .NE. 1) GO TO 601
000020            DING = SPAN(1)
000021            GO TO 650
000022        601 IF (NP .NE. NS) GO TO 602
000024            DING = SPAN(NS-1)
000026            GO TO 650
000026        602 DING = SPAN(NP) + SPAN(NP-1)
000032            GO TO 650
000032        600 IF (TYPE .NE. 4HMOMT) GO TO 610
            C        IMPACT DISTANCE FOR THE MOMENT INFLUENCE LINES
000034            DO 611 I = 2, NS
000035            IF (IOSP(I) .LT. NP) GO TO 611
000040            IF (DEAD .LT. 0.0) GO TO 612
000041            DING = SPAN(I-1)
000043            GO TO 650
000044        612 IF (NP .LT. (IOSP(I-1)+IOSP(I))/2) GO TO 625
000051            DING = (SPAN(I-1) + SPAN(I))/2.0
000056            GO TO 650
000056        625 DING = (SPAN(I-1) + SPAN(I-2))/2.0
000063            GO TO 650
000063        611 CONTINUE
000066        610 IF (TYPE .NE. 4HSHER) GO TO 620
            C        IMPACT DISTANCE FOR THE SHEAR INFLUENCE LINES
000070            DO 621 I = 2, NS
000071            IF (IOSP(I) .LT. NP) GO TO 621
000074            II = IOSP(I)
000075            JJ = IOSP(I-1)
000077            DING = COOR(II) - COOR(NP)
000104            DONG = COOR(NP) - COOR(JJ)
000110            IF (DONG .GT. DING) DING = DONG
000113            GO TO 650
000114        621 CONTINUE
            C        IMPACT DISTANCE FOR THE DEFLECTION INFLUENCE LINES
000117        620 DO 630 I = 2, NS
000121            IF (NP .GT. IOSP(I)) GO TO 630
000125            DING = SPAN(I-1)
000126            GO TO 650
000127        630 CONTINUE
000132            GO TO 650
000132       7000 DING = SPAN(1)
            C        IMPACT FACTOR CALCULATION
000134        650 HT = 50./((DING/12.) + 125.)
000137            IF (HT .GT. .3) HT = .3
000143            HT = HT + 1.0
000145            IMPACT = HT
000146            RETURN
000146            END
```

```
SUBPROGRAM LENGTH
000213

UNUSED COMPILER SPACE
042400



                   SUBROUTINE KEEP(NE,NOS,PLATH,PLWID,COVPL,CMIN,WMIN,IPT6)
          C           STORES THE ACCEPTABLE GIRDER
000013             COMMON/FOUR/WTOT,TCOST
000013             COMMON/A/ISAVE(80),SAVE(12,6,2),PLTSAV(80,2),PLWSAV(80,2)
000013             DIMENSION NOS(80),PLATH(80,2),PLWID(80,2),COVPL(12,6,2)
          C           CHECK IF THE GIRDER IS BETTER THAN THE PREVIOUS GIRDER
000013             IF (IPT6 .GT. .5) GO TO 7003
000017             IF (TCOST .LT. CMIN) GO TO 7004
000021             RETURN
000021        7003 IF (WTOT .LT. WMIN) GO TO 7004
000024             RETURN
          C           STORE THE ACCEPTABLE GIRDER
000025        7004 DO 7000 I = 1, NF
000027             DO 6999 J = 1, 2
000030             PLTSAV(I,J) = PLATH(I,J)
000036        6999 PLWSAV(I,J) = PLWID(I,J)
000043        7000 ISAVE(I) = NOS(I)
000047             CMIN = TCOST
000050             WMIN = WTOT
000051             DO 7001 I = 1, 12
000053             DO 7001 J = 1, 6
000054             DO 7001 K = 1, 2
000055        7001 SAVE(I,J,K) = COVPL(I,J,K)
000076             RETURN
000077             END

SUBPROGRAM LENGTH
000120


UNUSED COMPILER SPACE
042500



                   SUBROUTINE LOAD (TIL,TRLO,UNLO,SUBLEN,NA,IPT6,LENGTH,COOR,NP,TYPE,
                  1NS,IOSP,SPAN,STORE)
          C           LOADS A GIVEN INFLUENCE LINE
000021             DIMENSION TIL(81,3),TRLO(10),UNLO(9),RO(81),KOP(10),STORE(10),
                  1SUBLEN(80),IOSP(5),COOR(81),SPAN(4)
000021             INTEGER TYPE
000021             REAL LENGTH, ILINT, IMPACT
000021             COMMON/ONE/JPT3,IPT10
000021             MN = NS - 1
          C           ZERO OUT THE STORE ARRAY
000023             DO 811 I = 1, 10
```

```
000024      811 STORE(1) = 0.0
          C         SET UP THE STEEL SECTION INFLUENCE LINE
000030          DO 800 I = 1, NA
000031      800 RO(I) = TIL(I,1)
000040          CALL ILPROP(RO,SUBLEN,NA,KOP,ANEG,APOS,MAXOL,MINOL)
000047          IF (TYPE .NE. 4HSHER) GO TO 1850
000055          IF (MN .EQ. 1) GO TO 1850
000057          DO 1860 II = 2, MN
000060          IF (NP .NE. IOSP(II)) GO TO 1860
000063          IF (TIL(NP,1) .GT. 0.0) APOS = APOS - SUBLEN(NP)/2.0
000071          IF (TIL(NP,1) .LT. 0.0) ANEG = ANEG + SUBLEN(NP-1)/2.0
000077     1860 CONTINUE
000102     1850 ATOT = APOS + ANEG
000104          IF (JPT3 .LT. 9) GO TO 801
          C         DEAD LOAD REACTION FOR A NON - COMPOSITE SECTION
000107          UDL = TRLO(7) + TRLO(8) + TRLO(10)
000113          IF (UDL .LT. UNLO(4)+UNLO(5)+UNLO(9)) UDL =UNLO(4)+UNLO(5)+UNLO(9)
000125          STORE(1) = UDL * ATOT
000130          STORE(10) = 0.0
000131          GO TO 803
          C         DEAD LOAD AND LOW MODULUS REACTION FOR COMPOSITE ACTION SECTION
000131      801 UDL = TRLO(8) + TRLO(10)
000134          IF (UDL .LT. UNLO(5)+UNLO(9)) UDL = UNLO(5) + UNLO(9)
000142          STORE(1) = UDL * ATOT
000145          DO 802 I = 1, NA
000146      802 RO(I) = TIL(I,2)
000155          CALL ILPROP(RO,SUBLEN,NA,KOP,ANEG,APOS,MAXOL,MINOL)
000164          IF (TYPE .NE. 4HSHER) GO TO 1852
000172          IF (MN .EQ. 1) GO TO 1852
000174          DO 1861 II = 2, MN
000175          IF (NP .NE. IOSP(II)) GO TO 1861
000200          IF (TIL(NP,2) .GT. 0.0) APOS = APOS - SUBLEN(NP)/2.0
000206          IF (TIL(NP,2) .LT. 0.0) ANEG = ANEG + SUBLEN(NP-1)/2.0
000214     1861 CONTINUE
000217     1852 ATOT = APOS + ANEG
000221          ACA = UNLO(4)
000223          IF (ACA .LT. TRLO(7)) ACA = TRLO(7)
000227          STORE(10) = ATOT * ACA
          C         SET UP THE HIGH MODULUS INFLUENCE LINE IF REQUIRED
000232      803 IF (JPT3 .GT. 9) GO TO 804
000236          DO 805 I = 1, NA
000237      805 RO(I) = TIL(I,3)
000246          CALL ILPROP(RO,SUBLEN,NA,KOP,ANEG,APOS,MAXOL,MINOL)
000255          IF (TYPE .NE. 4HSHER) GO TO 804
000263          IF (MN .EQ. 1) GO TO 804
000265          DO 1862 II = 2, MN
000266          IF (NP .NE. IOSP(II)) GO TO 1862
000271          IF (TIL(NP,3) .GT. 0.0) APOS = APOS - SUBLEN(NP)/2.0
000277          IF (TIL(NP,3) .LT. 0.0) ANEG = ANEG + SUBLEN(NP-1)/2.0
000305     1862 CONTINUE
          C         BEGIN THE LIVE LOAD LOADINGS
000310      804 IF (IPT2 .EQ. 2) GO TO 830
000312          ACB = UNLO(3)
000313          IF (TYPE .EQ. 4HMOMT) ACB = UNLO(2)
          C         CALCULATE THE IMPACT FACTOR
000317          IF (UNLO(10) .LT. .1) GO TO 1800
000322          HT = 1.0
000324          GO TO 1801
```

```
000324      1800 HT = IMPACT(TYPE,NP,IOSP,NS,SPAN,STORE(1),COOR)                 LOAD  70
          C          CALCULATE THE POSITIVE AND NEGATIVE LANE LOADINGS           LOAD  71
000340      1801 STORE(2) = HT * (UNLO(1) * APOS + ACR * RO(MAXOL))              LOAD  72
000353           SPEK = 0.                                                       LOAD  73
000354           IF (TYPE .NE. 4HMONT) GO TO 812                                 LOAD  74
000356           IF (MN .EQ. 1) GO TO 812                                        LOAD  75
000360           DO 806 J = 1, MN                                                LOAD  76
000361           IF (MINOL .LT. IOSP(J)) GO TO 806                               LOAD  77
000364           IF (MINOL .GT. IOSP(J+1)) GO TO 806                             LOAD  78
000367           IF (J .NE. 1) GO TO 807                                         LOAD  79
000370           L = IOSP(2)                                                     LOAD  80
000372           M = IOSP(3)                                                     LOAD  81
000373           DO 808 K = L, M                                                 LOAD  82
000375       808 IF (RO(K) .LT. SPEK) SPEK = RO(K)                               LOAD  83
000405           GO TO 812                                                       LOAD  84
000405       807 IF (J .NE. MN) GO TO 809                                        LOAD  85
000407           L = IOSP(MN - 1)                                                LOAD  86
000412           M = IOSP(MN)                                                    LOAD  87
000414           DO 813 K = L, M                                                 LOAD  88
000416       813 IF (RO(K) .LT. SPEK) SPEK = RO(K)                               LOAD  89
000426           GO TO 812                                                       LOAD  90
000426       809 L = IOSP(J-1)                                                   LOAD  91
000431           M = IOSP(J)                                                     LOAD  92
000433           DO 814 K = L, M                                                 LOAD  93
000435       814 IF (RO(K) .LT. SPEK) SPEK = RO(K)                               LOAD  94
000445           L = IOSP(J+1)                                                   LOAD  95
000447           M = IOSP(J+2)                                                   LOAD  96
000451           DO 815 K = L, M                                                 LOAD  97
000453       815 IF (RO(K) .LT. SPEK) SPEK = RO(K)                               LOAD  98
000463           GO TO 812                                                       LOAD  99
000463       806 CONTINUE                                                        LOAD 100
000466       812 SPEK = RO(MINOL) + SPEK                                         LOAD 101
000471           STORE(3) = HT * (UNLO(1) * ANEG + ACR * SPEK)                   LOAD 102
          C          CALCULATE THE SIDEWALK LOADINGS                             LOAD 103
000477           STORE(8) = APOS * UNLO(7)                                       LOAD 104
000502           STORE(9) = ANEG * UNLO(7)                                       LOAD 105
          C          INTERSTATE LOADINGS ( POS AND NEG BASED ON A 4 FOOT SPACING) LOAD 106
000504           RR = ILINT(MAXOL,RO,COOR,1,48.,LENGTH,SUBLEN)                   LOAD 107
000520           PR = RR + RO(MAXOL)                                             LOAD 108
000523           RR = ILINT(MAXOL,RO,COOR,2,48.,LENGTH,SUBLEN)                   LOAD 109
000536           SR = RR + RO(MAXOL)                                             LOAD 110
000541           IF (PR .LT. SR) PR = SR                                         LOAD 111
000544           RR = ILINT(MAXOL,RO,COOR,1,24.,LENGTH,SUBLEN)                   LOAD 112
000560           TR = ILINT(MAXOL,RO,COOR,2,24.,LENGTH,SUBLEN)                   LOAD 113
000574           SR = RR + TR                                                    LOAD 114
000576           IF (PR .LT. SR) PR = SR                                         LOAD 115
000601           STORE(4) = HT * UNLO(6) * PR                                    LOAD 116
000605           RR = ILINT(MINOL,RO,COOR,1,48.,LENGTH,SUBLEN)                   LOAD 117
000620           PR = RR + RO(MINOL)                                             LOAD 118
000623           RR = ILINT(MINOL,RO,COOR,2,48.,LENGTH,SUBLEN)                   LOAD 119
000636           SR = RR + RO(MINOL)                                             LOAD 120
000641           IF (PR .GT. SR) PR = SR                                         LOAD 121
000644           RR = ILINT(MINOL,RO,COOR,1,24.,LENGTH,SUBLEN)                   LOAD 122
000660           TR = ILINT(MINOL,RO,COOR,2,24.,LENGTH,SUBLEN)                   LOAD 123
000674           SR = RR + TR                                                    LOAD 124
000676           IF (PR .GT. SR) PR = SR                                         LOAD 125
000701           STORE(5) = HT * UNLO(6) * PR                                    LOAD 126
000705       830 IF (IPT2 .EQ. 1) GO TO 870                                      LOAD 127
```

```
       C          START THE TRUCK LOADING
000707          POS = 0.0
000710          SEG = 0.0
000711          PFRA = .4
000712          OCA = TRLO(5) + TRLO(4)
000715          OCB = TRLO(6) + TRLO(4)
       C          MOVE THE TRUCK FROM LEFT TO RIGHT
000717          J = 1
       C          REAR AXLE ON THE POINT
       C          CALCULATE FOR THE MINIMUM SPACING
000720      834 RAL = TRLO(3) * RO(J)
000723          RR = ILINT(J,RO,COOR,1,TRLO(5),LENGTH,SUBLEN)
000741          SR = ILINT(J,RO,COOR,1,OCA,LENGTH,SUBLEN)
000755          TTL = RAL + TRLO(2) * RR + TRLO(1) * SR
000763          IF (TTL .GT. POS) POS = TTL
000766          IF (TTL .LT. SEG) SEG = TTL
       C          CALCULATE FOR THE MAXIMUM SPACING
000771          RR = ILINT(J,RO,COOR,1,TRLO(6),LENGTH,SUBLEN)
001007          SR = ILINT(J,RO,COOR,1,OCB,LENGTH,SUBLEN)
001023          TTL = RAL + TRLO(2) * RR + TRLO(1) * SR
001031          IF (TTL .GT. POS) POS = TTL
001034          IF (TTL .LT. SEG) SEG = TTL
       C          FRONT AXLE ON THE POINT
001037          FAL = RO(J) * TRLO(1)
001042          RR = ILINT(J,RO,COOR,2,TRLO(4),LENGTH,SUBLEN)
001057          PAL = FAL + RR * TRLO(2)
001062          IF (PAL .GT. POS) POS = PAL
001065          IF (PAL .LT. SEG) SEG = PAL
001070          IF (COOR(J) - OCA .LT. 0.0) GO TO 835
001074          IF (PAL .LT. PFRA * POS) GO TO 831
001100          GO TO 832
001100      831 IF (PAL .GT. PFRA * SEG) GO TO 835
       C          START WIGGLING THE BACK AXLE
001105      832 RR = ILINT(J,RO,COOR,2,OCA,LENGTH,SUBLEN)
001121          TTL = PAL + RR * TRLO(3)
001124          IF (TTL .GT. POS) POS = TTL
001127          IF (TTL .LT. SEG) SEG = TTL
001132          SR = ILINT(J,RO,COOR,2,OCB,LENGTH,SUBLEN)
001146          TTL = PAL + SR * TRLO(3)
001151          IF (TTL .GT. POS) POS = TTL
001154          IF (TTL .LT. SEG) SEG = TTL
001157          K = J + 1
001161      833 IF (K .EQ. 1) GO TO 835
001163          K = K - 1
001164          IF (COOR(K) .GT. COOR(J) - OCA) GO TO 833
001173          IF (COOR(K) .LT. COOR(J) - OCB) GO TO 835
001201          TTL = PAL + RO(K) * TRLO(3)
001205          IF (TTL .GT. POS) POS = TTL
001210          IF (TTL .LT. SEG) SEG = TTL
001213          GO TO 833
       C          PLACE THE SECOND AXLE ON THE POINT
001214      835 CAL = TRLO(2) * RO(J)
001217          RR = ILINT(J,RO,COOR,1,TRLO(4),LENGTH,SUBLEN)
001235          PAL = CAL + RR * TRLO(1)
001240          IF (PAL .GT. POS) POS = PAL
001243          IF (PAL .LT. SEG) SEG = PAL
001246          IF (COOR(J) - TRLO(5) .LT. 0.0) GO TO 840
001253          IF (PAL .LT. PFRA * POS) GO TO 836
```

```
001256          GO TO 837                                                  LOAD 186
001257      836 IF (PAL .GT. PERA * SEG) GO TO 840                         LOAD 187
         C          START WIGGLING THE BACK AXLE                           LOAD 188
001264      837 RR = ILINT(J,RO,COOR,2,TRLO(5),LENGTH,SUBLEN)              LOAD 189
001302          TTL = PAL + RR * TRLO(3)                                   LOAD 190
001305          IF (TTL .GT. POS) POS = TTL                                LOAD 191
001310          IF (TTL .LT. SEG) SEG = TTL                                LOAD 192
001313          SR = ILINT(J,RO,COOR,2,TRLO(6),LENGTH,SUBLEN)              LOAD 193
001331          TTL = PAL + SR * TRLO(3)                                   LOAD 194
001334          IF (TTL .GT. POS) POS = TTL                                LOAD 195
001337          IF (TTL .LT. SEG) SEG = TTL                                LOAD 196
001342          K = J + 1                                                  LOAD 197
001344      838 IF (K .EQ. 1) GO TO 840                                    LOAD 198
001346          K = K - 1                                                  LOAD 199
001347          IF (COOR(K) .GT. COOR(J) - OCA) GO TO 838                  LOAD 200
001356          IF (COOR(K) .LT. COOR(J) - OCB) GO TO 840                  LOAD 201
001364          TTL = PAL + RO(K) * TRLO(3)                                LOAD 202
001370          IF (TTL .GT. POS) POS = TTL                                LOAD 203
001373          IF (TTL .LT. SEG) SEG = TTL                                LOAD 204
001376          GO TO 838                                                  LOAD 205
001377      840 J = J + 1                                                  LOAD 206
001401          IF (J .LE. NA) GO TO 834                                   LOAD 207
         C          MOVE THE TRUCK FROM THE RIGHT TO THE LEFT              LOAD 208
001403          J = NA                                                     LOAD 209
         C          REAR AXLE ON THE POINT                                 LOAD 210
         C          CALCULATE FOR THE MINIMUM SPACING                      LOAD 211
001403      841 RAL = TRLO(3) * RO(J)                                      LOAD 212
001406          RR = ILINT(J,RO,COOR,2,TRLO(5),LENGTH,SUBLEN)              LOAD 213
001424          SR = ILINT(J,RO,COOR,2,OCA,LENGTH,SUBLEN)                  LOAD 214
001440          TTL = RAL + TRLO(2) * RR + TRLO(1) * SR                    LOAD 215
001446          IF (TTL .GT. POS) POS = TTL                                LOAD 216
001451          IF (TTL .LT. SEG) SEG = TTL                                LOAD 217
         C         .CALCULATE FOR THE MAXIMUM SPACING                      LOAD 218
001454          RR = ILINT(J,RO,COOR,2,TRLO(6),LENGTH,SUBLEN)              LOAD 219
001472          SR = ILINT(J,RO,COOR,2,OCB,LENGTH,SUBLEN)                  LOAD 220
001506          TTL = RAL + TRLO(2) * RR + TRLO(1) * SR                    LOAD 221
001514          IF (TTL .GT. POS) POS = TTL                                LOAD 222
001517          IF (TTL .LT. SEG) SEG = TTL                                LOAD 223
         C          FRONT AXLE ON THE POINT                                LOAD 224
001522          FAL = RO(J) * TRLO(1)                                      LOAD 225
001525          RR = ILINT(J,RO,COOR,1,TRLO(4),LENGTH,SUBLEN)              LOAD 226
001542          PAL = FAL + RR * TRLO(2)                                   LOAD 227
001545          IF (PAL .GT. POS) POS = PAL                                LOAD 228
001550          IF (PAL .LT. SEG) SEG = PAL                                LOAD 229
001553          IF (COOR(J) + OCA .GT. LENGTH) GO TO 845                   LOAD 230
001562          IF (PAL .LT. PERA * POS) GO TO 842                         LOAD 231
001565          GO TO 843                                                  LOAD 232
001566      842 IF (PAL .GT. PERA * SEG) GO TO 845                         LOAD 233
         C          START WIGGLING THE BACK AXLE                           LOAD 234
001573      843 RR = ILINT(J,RO,COOR,1,OCA,LENGTH,SUBLEN)                  LOAD 235
001607          TTL = PAL + RR * TRLO(3)                                   LOAD 236
001612          IF (TTL .GT. POS) POS = TTL                                LOAD 237
001615          IF (TTL .LT. SEG) SEG = TTL                                LOAD 238
001620          SR = ILINT(J,RO,COOR,1,OCB,LENGTH,SUBLEN)                  LOAD 239
001634          TTL = PAL + SR * TRLO(3)                                   LOAD 240
001637          IF (TTL .GT. POS) POS = TTL                                LOAD 241
001642          IF (TTL .LT. SEG) SEG = TTL                                LOAD 242
001645          DO 844 K = J, NA                                           LOAD 243
```

```
001647          IF (COOR(K) .LT. COOR(J) + DCA) GO TO 844                          LOAD 244
001655          IF (COOR(K) .GT. COOR(J) + DCB) GO TO 845                          LOAD 245
001665          TTL = PAL + RO(K) * TRLO(3)                                         LOAD 246
001670          IF (TTL .GT. POS) POS = TTL                                         LOAD 247
001673          IF (TTL .LT. SEG) SEG = TTL                                         LOAD 248
001676      844 CONTINUE                                                            LOAD 249
          C        PLACE THE SECOND AXLE ON THE POINT                               LOAD 250
001701      845 CAL = TRLO(2) * RO(J)                                               LOAD 251
001704          RR = ILINT(J,RO,COOR,2,TRLO(4),LENGTH,SUBLEN)                       LOAD 252
001722          PAL = CAL + RR * TRLO(1)                                            LOAD 253
001725          IF (PAL .GT. POS) POS = PAL                                         LOAD 254
001730          IF (PAL .LT. SEG) SEG = PAL                                         LOAD 255
001733          IF (COOR(J) + TRLO(5) .GT. LENGTH) GO TO 850                        LOAD 256
001742          IF (PAL .LT. PERA * POS) GO TO 846                                  LOAD 257
001745          GO TO 848                                                           LOAD 258
001746      846 IF (PAL .GT. PERA * SEG) GO TO 850                                  LOAD 259
          C        START WIGGLING THE BACK AXLE                                     LOAD 260
001753      848 RR = ILINT(J,RO,COOR,1,TRLO(5),LENGTH,SUBLEN)                       LOAD 261
001771          TTL = PAL + RR * TRLO(3)                                            LOAD 262
001774          IF (TTL .GT. POS) POS = TTL                                         LOAD 263
001777          IF (TTL .LT. SEG) SEG = TTL                                         LOAD 264
002002          SR = ILINT(J,RO,COOR,1,TRLO(6),LENGTH,SUBLEN)                       LOAD 265
002020          TTL = PAL + SR * TRLO(3)                                            LOAD 266
002023          IF (TTL .GT. POS) POS = TTL                                         LOAD 267
002026          IF (TTL .LT. SEG) SEG = TTL                                         LOAD 268
002031          DO 847 K = J, NA                                                    LOAD 269
002033          IF (COOR(K) .LT. COOR(J) + DCA) GO TO 847                          LOAD 270
002041          IF (COOR(K) .GT. COOR(J) + DCB) GO TO 850                          LOAD 271
002051          TTL = PAL + RO(K) * TRLO(3)                                         LOAD 272
002054          IF (TTL .GT. POS) POS = TTL                                         LOAD 273
002057          IF (TTL .LT. SEG) SEG = TTL                                         LOAD 274
002062      847 CONTINUE                                                            LOAD 275
002065      850 J = J - 1                                                           LOAD 276
002067          IF (J .GE. 1) GO TO 841                                             LOAD 277
          C        CALCULATE THE IMPACT FACTOR                                      LOAD 278
002070          IF (TRLO(9) .LT. .1) GO TO 1802                                     LOAD 279
002073          HT = 1.0                                                            LOAD 280
          C        STORE THE MAXIMUM POSITIVE AND NEGATIVE TRUCKLOADS               LOAD 281
002075     1802 STORE(6) = HT * POS                                                 LOAD 282
002100          STORE(7) = HT * SEG                                                 LOAD 283
002102      870 CONTINUE                                                            LOAD 284
002102          RETURN                                                              LOAD 285
002103          END                                                                 LOAD 286

SUBPROGRAM LENGTH
002367

UNUSED COMPILER SPACE
034500



                SUBROUTINE OBJECT                                                   OBJT   1
          C        DETERMINES THE WEIGHT AND/OR COST OF A DESIGN                    OBJT  1A
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWIDTH(18),FLTH(18),WEBTH(18), OBJT   2
```

```
                 1DAP(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),
                 2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,
                 3SLABWD,SLABTH,MR,COOR(81),RTL(5,8,3),CC(5,5),FY,SPAN(4),IOSP(5),
                 4TMOEI(80),NA,LENGTH,RD(81),SUMLEN(80),NS,UNLD(9),IPLD(10),PROP(3),
                 5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),BEAR(5,3),
                 6COST(9),CONSP(16,3,4),COVPL(12,6,2)
000002            COMMON/ONE/JPT3,IPT10
000002            COMMON/THREE/AS,NCS,IPT6
000002            COMMON/FOUR/WTOT,ICOST
000002            COMMON/EIGHT/IST
000002            REAL LENGTH
           C          DETERMINE WEIGHT AND/OR COST OF ROLLED SECTIONS
000002            MN = NS - 1
000004            WRS = LENGTH * SAREA(IST) * 490./1728.
000007            IF (IPT6 .EQ. 1) GO TO 6000
000011            CRS = WRS * COST(1)
000013       6000 CONTINUE
           C          DETERMINE WEIGHT AND/OR COST OF COVER PLATES
000013       6005 CPW = 0.0
000014            CPC = 0.0
000015            DO 6006 K = 1, 2
000016            DO 6007 I = 1, 12
000017            IF (ABS(COVPL(I,1,K)).LT..0001.AND.ABS(COVPL(I,2,K)).LT..0001)
                 1GO TO 6006
000037            CPL = COVPL(I,2,K) - COVPL(I,1,K)
000042            CPA = COVPL(I,3,K) * COVPL(I,4,K)
000045            PARW = CPA * CPL * 490./1728.
000047            CPW = CPW + PARW
000051            IF (IPT6 .EQ. 1) GO TO 6007
000053            CPC = CPC + PARW * COST(2)
000055       6007 CONTINUE
000057       6006 CONTINUE
           C          DETERMINE WEIGHT AND/OR COST OF BEARING STIFFENERS
000061            WBS = 0.0
000062            CBS = 0.0
000063            CBSW = 0.0
000064            DO 6010 I = 1, NS
000065            IF (ABS(BEAR(I,1)) .LT. .0001) GO TO 6010
000071            IA = IOSP(I)
000072            IF (I .EQ. NS) IA = IOSP(I) - 1
000076            IA = NOS(IA)
000100            BH = DEPTH(IA) - 2. * FLTH(IA)
000103            BV = 2. * BH * BEAR(I,1) * BEAR(I,2)
000107            PARW = BV * 490./1728.
000111            WBS = WBS + PARW
000112            IF (IPT6 .EQ. 1) GO TO 6010
000114            CBS = CBS + COST(3) * PARW
000117            WV = 2. * BH * BEAR(I,3) * BEAR(I,3)
000122            CBSW = CBSW + 4. * COST(6) + WV * COST(7) + 2. * BH * COST(8)
000131       6010 CONTINUE
           C          DETERMINE WEIGHT AND/OR COST OF THE SHEAR CONNECTORS
000134            WCON = 0.0
000135            CCON = 0.0
000136            IF (JPT3 .GT. 9) GO TO 6015
000141            NROW = MN
000142            DO 6012 K = 1, MN
000143            DO 6013 I = 1, 16
000144            IF (ABS(CONSP(I,1,K)) .LT. .0001) GO TO 6012
```

```
000152          NRA = ((CONSP(I,3,K) - CONSP(I,2,K)) / CONSP(I,1,K)                      OBJT  61
000161          IF (ABS(NRA*CONSP(I,1,K)-CONSP(I,3,K)+CONSP(I,2,K)) .GT. .0001)          OBJT  62
               1 NRA = NRA + 1                                                           OBJT  63
000171          NROW = NROW + NRA                                                        OBJT  64
000173     6013 CONTINUE                                                                 OBJT  65
000175     6012 CONTINUE                                                                 OBJT  66
000200          NCONN = NROW * NCS                                                       OBJT  67
000202          WCON = NCONN * .77561493                                                 OBJT  68
000205          IF (IPT6 .EQ. 1) GO TO 6015                                              OBJT  69
000207          CCON = WCON * COST(4) + NCONN * COST(5)                                  OBJT  70
000213     6015 WTOT = WRS + CPW + WBS + WCON                                            OBJT  71
000217          IF (IPT6 .NE. 1) GO TO 6050                                              OBJT  72
000221          RETURN                                                                   OBJT  73
         C      DETERMINE THE COST OF THE WELDS                                          OBJT  74
000222     6050 CWELD = CHSW                                                             OBJT  75
         C         DETERMINE THE COST OF THE COVER PLATE WELDS                           OBJT  76
000224          CPWC = 0.0                                                               OBJT  77
000225          DO 6020 K = 1, 2                                                         OBJT  78
000226          IA = 0                                                                   OBJT  79
000227          DO 6021 I = 1, 12                                                        OBJT  80
000231          IF (ABS(COVPL(I,1,K)) .LT. .0001 .AND. ABS(COVPL(I,2,K)) .LT.            OBJT  81
               1.0001) GO TO 6022                                                        OBJT  82
000251          IA = I                                                                   OBJT  83
000252     6021 CONTINUE                                                                 OBJT  84
000254     6022 IF (IA .EQ. 0) GO TO 6020                                                OBJT  85
000255          DO 6023 I = 1, IA                                                        OBJT  86
000257          CW = 0.0                                                                 OBJT  87
000260          IF (I .EQ. 1) GO TO 6025                                                 OBJT  88
000262          IF (ABS(COVPL(I,1,K)-COVPL(I-1,2,K)) .LT. .0001) GO TO 6024              OBJT  89
000272     6025 WLEN = COVPL(I,4,K) * 4.                                                 OBJT  90
000277          WV = WLEN * COVPL(I,5,K) * COVPL(I,5,K)/2.                               OBJT  91
000303          CW = CW + WLEN*COST(8) + WV*COST(7) + COST(6)                            OBJT  92
000312          GO TO 6026                                                               OBJT  93
000312     6024 TH = COVPL(I,3,K)                                                        OBJT  94
000316          IF (COVPL(I-1,3,K) .LT. COVPL(I,3,K)) TH = COVPL(I-1,3,K)                OBJT  95
000326          WID = COVPL(I,4,K)                                                       OBJT  96
000332          IF (COVPL(I-1,4,K) .LT. COVPL(I,4,K)) WID = COVPL(I-1,4,K)               OBJT  97
000341          WV = (.125*TH + .288675*TH**2) * WID                                     OBJT  98
000346          BWC = WV*COST(7) + WID*COST(8) + COST(6)                                 OBJT  99
000352          CW = CW + BWC                                                            OBJT 100
000354     6026 IF (I .EQ. IA) GO TO 6027                                                OBJT 101
000356          IF (ABS(COVPL(I+1,1,K) - COVPL(I,2,K)) .LT. .0001) GO TO 6029            OBJT 102
000366     6027 WLEN = COVPL(I,4,K) * 4.                                                 OBJT 103
000373          WV = WLEN * COVPL(I,5,K) * COVPL(I,5,K)/2.                               OBJT 104
000377          CW = CW + WLEN*COST(8) + WV*COST(7) + COST(6)                            OBJT 105
000406     6029 SWL = COVPL(I,2,K) - COVPL(I,1,K)                                        OBJT 106
000415          IF (I .EQ. 1 .OR. I .EQ. IA) GO TO 6030                                  OBJT 107
000424          IF (ABS(COVPL(I+1,1,K) - COVPL(I,2,K)) .GT. .0001)                       OBJT 108
               1SWL = SWL - 1.5 * COVPL(I,4,K)                                           OBJT 109
000436          IF (ABS(COVPL(I,1,K) - COVPL(I-1,2,K)) .GT. .0001)                       OBJT 110
               1SWL = SWL - 1.5 * COVPL(I,4,K)                                           OBJT 111
000451          GO TO 6031                                                               OBJT 112
000452     6030 SWL = SWL - 1.5 * COVPL(I,4,K)                                           OBJT 113
000460          IF(I.EQ.1.AND.ABS(COVPL(2,1,K)-COVPL(I,2,K)).GT..0001)                   OBJT 114
               1SWL = SWL - 1.5 * COVPL(I,4,K)                                           OBJT 115
000501          IF (I.EQ.IA.AND.ABS(COVPL(I,1,K)-COVPL(I-1,2,K)) .GT. .0001)             OBJT 116
               1SWL = SWL - 1.5 * COVPL(I,4,K)                                           OBJT 117
000524     6031 WV = SWL * COVPL(I,6,K) * COVPL(I,6,K)/2.                                OBJT 118
```

```
000532          CSLW = WV * COST(7) + SWL * COST(8) + COST(6)              OBJT 119
000536          CW = CW + 2. * CSLW                                        OBJT 120
000541          CPWC = CPWC + CW                                           OBJT 121
000542     6023 CONTINUE                                                   OBJT 122
000545     6020 CONTINUE                                                   OBJT 123
000547          CWELD = CWELD + CPWC                                       OBJT 124
000551          TCOST = CRS + CPC + CHS + CCON + CWELD                     OBJT 125
000556          RETURN                                                     OBJT 126
000557          END                                                        OBJT 127

SUBPROGRAM LENGTH
000677

UNUSED COMPILER SPACE
040100



                SUBROUTINE PRCON(NS,CONSP,NCS)                             PRCO   1
          C        PRINT THE CONNECTOR INFORMATION                         PRCO   2
000006          COMMON/ONE/JP13,IPT10                                      PRCO   3
000006          DIMENSION CONSP(16,3,4)                                    PRCO   4
000006          MN = NS - 1                                                PRCO   5
          C        ONLY PRINT TABLE IN DESIGN IS COMPOSITE                 PRCO   6
000010          IF (JP13 .GT. 9) GO TO 5000                                PRCO   7
          C        PRINT THE HEADING                                       PRCO   8
000013          WRITE (6,4190)                                             PRCO   9
000016     4190 FORMAT (1H0,10X,31HSHEAR CONNECTOR INFORMATION...//20X,    PRCO  10
               137HPLACE THE SHEAR CONNECTORS AS FOLLOWS/)                 PRCO  11
          C        PRINT THE CONNECTOR INFORMATION FOR EACH SPAN           PRCO  12
000016          DO 4191 I = 1, MN                                          PRCO  13
000022          WRITE (6,4194) I                                           PRCO  14
000027     4194 FORMAT (20X,16HFOR SPAN NUMBER ,I1,3H...)                  PRCO  15
000027          DO 4192 J = 1, 16                                          PRCO  16
000033          IF (ABS(CONSP(J,1,I)) .LT. .0001) GO TO 4191               PRCO  17
000041          XD = CONSP(J,3,I) - CONSP(J,2,I)                           PRCO  18
000047          NXD = XD / CONSP(J,1,I)                                    PRCO  19
000052          IF (ABS(NXD*CONSP(J,1,I) - XD) .GT. .0001) NXD = NXD + 1   PRCO  20
000061     4192 WRITE (6,4193) CONSP(J,2,I), NXD, CONSP(J,1,I), XD         PRCO  21
000111     4193 FORMAT (25X,12HSTARTING AT ,F9.2,6H, USE ,I3,11H SPACES AT ,F4.1, PRCO  22
               116H INCHES/SPACE = ,F10.2,8H INCHES.)                      PRCO  23
000111     4191 CONTINUE                                                   PRCO  24
          C        PRINT THE NUMBER OF CONNECTORS PER SECTION              PRCO  25
000114          WRITE (6,4195) NCS                                         PRCO  26
000121     4195 FORMAT (1H0,20X,4HUSE ,I1,24H CONNECTORS IN EACH ROW.)     PRCO  27
000121          GO TO 5001                                                 PRCO  28
          C        PRINT THE NON-COMPOSITE DESIGN STATEMENT                PRCO  29
000124     5000 WRITE (6,5002)                                             PRCO  30
000130     5002 FORMAT (1H0///10X,75HTHERE ARE NO SHEAR CONNECTORS REQUIRED BECAUSPRCO  31
               1E THE DESIGN IS NON-COMPOSITE.//)                          PRCO  32
000130     5001 RETURN                                                     PRCO  33
000131          END                                                        PRCO  34

SUBPROGRAM LENGTH
```

```
000235

UNUSED COMPILER SPACE
042300



                    SUBROUTINE PRCOV (COVPL)
               C          PRINT THE COVER PLATE INFORMATION
000003              COMMON/ONE/JPT3,IPT10
000003              DIMENSION COVPL(12,6,2)
               C          PRINT THE HEADING
000003              WRITE (6,2000)
000007          2000 FORMAT (1H0,26HCOVER PLATE INFORMATION...//)
               C          BEGIN THE LOOP FOR TOP AND BOTTOM PLATES
000007              DO 2001 K = 1, 2
000012              IF (K .EQ. 1) WRITE (6,2002)
000020              IF (K .EQ. 2) WRITE (6,2003)
000027          2002 FORMAT (5X,30HON THE TOP OF THE GIRDER USE../)
000027          2003 FORMAT (5X,33HON THE BOTTOM OF THE GIRDER USE../)
000027              IF (ABS(COVPL(1,1,K)).LT..0001.AND.ABS(COVPL(1,2,K)).LT..0001)
                   1GO TO 2005
               C          BEGIN THE LOOP FOR EACH PLATE
000045              DO 2004 I = 1, 12
000046              IF (ABS(COVPL(I,1,K)).LT..0001.AND.ABS(COVPL(I,2,K)).LT..0001)
                   1GO TO 2001
               C          PRINT THE INFORMATION FOR EACH PLATE
000067              WRITE (6,2006) COVPL(I,4,K), COVPL(I,3,K), COVPL(I,1,K),
                   1COVPL(I,2,K), COVPL(I,5,K), COVPL(I,6,K)
000122          2006 FORMAT(10X,4HUSE ,F6.2,3H X ,F6.3,17H INCH PLATE FROM ,
                   1F8.1,4H TO ,F8.1,6H WITH ,F6.3,17H END WELDS AND A ,
                   2F6.4,11H SEAL WELD.)
000122          2004 CONTINUE
000125              GO TO 2001
               C          PRINT THE NO PLATE STATEMENT
000126          2005 WRITE (6,2010)
000132          2010 FORMAT (1H0,10X,25HNO COVER PLATES REQUIRED.//)
000132          2001 CONTINUE
               C          PRINT THE NON-COMPOSITE STATEMENT
000135              IF (JPT3 .GT. 9) WRITE (6,2200)
000144          2200 FORMAT (1H0,10X,28HTHE DESIGN IS NON-COMPOSITE.)
000144              RETURN
000145              END


SUBPROGRAM LENGTH
000247

UNUSED COMPILER SPACE
042300


                    SUBROUTINE PRDEF (ND,D,NS,S,ID,IPT2,USLL)
```

```
          C          PRINT THE DEFLECTION INFORMATION                                PHDE     2
000012           COMMON/ONE/JPT3,IPT10                                               PHDE     3
000012           DIMENSION D(20,4), S(4), ID(20)                                     PHDE     4
000012           MN = NS - 1                                                         PHDE     5
000014           C = 600.                                                            PHDE     6
000015           IF (IPT2 .EQ. 2) GO TO 250                                          PHDE     7
000017           IF (ABS(USLL) .GT. .00001) C = 1000.                                PHDE     8
          C          PRINT THE DEFLECTION TABLE                                      PHDE     9
000024       250 WRITE (6,210)                                                       PHDE    10
000030       210 FORMAT (1H0,23HGIRDER DEFLECTIONS.....//10X,58HTHE GIRDER DEFLECTIPHDE    11
                1ONS ARE SHOWN IN THE FOLLOWING TABLE...//8X,9HLIVE LOAD,20X,        PHDE    12
                22HPT,10X,9HDEAD LOAD,10X,12HSUPERIMPOSED,10X,8HDOWNWARD,10X,        PHDE    13
                36HUPWARD//)                                                         PHDE    14
000030           DO 215 I = 1, ND                                                    PHDE    15
000035           IA = ID(I)                                                          PHDE    16
000037       215 WRITE (6,211) IA, D(I,1), D(I,3), D(I,2), D(I,4)                    PHDE    17
000103       211 FORMAT (20X,I2,10X,F7.4,14X,F7.4,13X,F7.4,1 X,F .4)                 PHDE    18
000103           WRITE (6,212)                                                       PHDE    19
000106       212 FORMAT (20X, 36HTHE ABOVE DEFLECTIONS ARE IN INCHES.)               PHDE    20
          C          PRINT THE MAXIMUM ALLOWABLE LIVE LOAD DEFLECTIONS               PHDE    21
000106           WRITE (6,231)                                                       PHDE    22
000112       231 FORMAT (1H0)                                                        PHDE    23
000112           DO 230 I = 1, MN                                                    PHDE    24
000117           DM = S(I) / C                                                       PHDE    25
000122       230 WRITE (6,235) I, DM                                                 PHDE    26
000137       235 FORMAT (11X,51HTHE MAXIMUM ALLOWABLE LIVE LOAD DEFLECTION IN SPAN PHDE    27
                1,I1,3H = ,F7.4,8H INCHES.)                                          PHDE    28
000137           IF (JPT3 .GT. 9) WRITE (6,240)                                      PHDE    29
000151       240 FORMAT (1H0,10X,74HTHE SUPERIMPOSED DEFLECTIONS ARE ZERO BECAUSE TPHDE    30
                1HE DESIGN IS NON-COMPOSITE.)                                        PHDE    31
000151           RETURN                                                              PHDE    32
000152           END                                                                 PHDE    33

SUBPROGRAM LENGTH
000263

UNUSED COMPILER SPACE
042200


                 SUBROUTINE PRSEC(NF,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CL,        PHSE     1
                1IHCON,CH,COOR,SUBLEN,KEY,NCY)                                       PHSE     2
          C          PRINTS THE SECTION PROPERTIES FOR A DESIGN                      PHSE     3
000022           COMMON/ONE/JPT3,IPT10                                               PHSE     4
000022           DIMENSION NOS(80),NAME(18),PLWID(80,2),PLATH(80,2),ISTEEL(80),      PHSE     5
                1CS(80),ILCON(80),CL(80),IHCON(80),CH(80),COOR(81),SUBLEN(80)        PHSE     6
000022           REAL ILCON,IHCON,ISTEEL                                             PHSE     7
          C          DETERMINE THE HEADING FOR THE TABLE                             PHSE     8
000022           IF (KEY .EQ. 0) GO TO 160                                           PHSE     9
000023           IF (KEY .EQ. 2) GO TO 60                                            PHSE    10
000025           WRITE (6,170) NCY                                                   PHSE    11
000032       170 FORMAT (1H1///40X,35HTHE OPTIMUM DESIGN FOR CYCLE NUMBER ,I2//)     PHSE    12
000032           GO TO 180                                                           PHSE    13
000036        60 WRITE (6,61)                                                        PHSE    14
```

```
000042      61 FORMAT (1H1///50X,15HTHE FINAL DESIGN//)                          PRSE  15
000042         GO TO 180                                                         PRSE  16
000046     160 WRITE (6,161)                                                     PRSE  17
000052     161 FORMAT (1H1)                                                      PRSE  18
000052     180 IF (JPT3 .GT. 9) GO TO 1000                                       PRSE  19
         C           PRINTS THE TABLE HEADING FOR A COMPOSITE DESIGN             PRSE  20
000061         WRITE (6,151)                                                     PRSE  21
000064     151 FORMAT (1H0,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HBOT,4X,3HBOT,        PRSE  22
              156X,4HLEFT,4X,5HRIGHT/2X,1HI,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,  PRSE  23
              25HPLWID,2X,5HPLATH,2X,6HISTEFL,5X,2HCS,5X,5HTLCON,5X,2HCL,5X,     PRSE  24
              35HTHCON,5X,2HCH,6X,4HCOOR,5X,4HCOOR,3X,6HSUBLEN//)                PRSE  25
000064         GO TO 1001                                                        PRSE  26
         C           PRINT THE TABLE HEADING FOR A NON-COMPOSITE DESIGN          PRSE  27
000070    1000 WRITE (6,1151)                                                    PRSE  28
000074    1151 FORMAT (1H0,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HBOT,4X,3HBOT,        PRSE  29
              122X,4HLEFT,4X,5HRIGHT/2X,1HI,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,  PRSE  30
              25HPLWID,2X,5HPLATH,2X,6HISTEFL,5X,2HCS,6X,                        PRSE  31
              3                      4HCOOR,5X,4HCOOR,3X,6HSUBLEN//)             PRSE  32
         C           PRINTS THE TABLE FOR A COMPOSITE DESIGN                     PRSE  33
000074    1001 DO 155 J = 1, NE                                                  PRSE  34
000101         K = LOS(J)                                                        PRSE  35
000103         IF (JPT3 .GT. 9) GO TO 1155                                       PRSE  36
000107         WRITE (6,152) J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(J,2),         PRSE  37
              1PLATH(J,2),ISTEFL(J),CS(J),TLCON(J),CL(J),THCON(J),CH(J),COOR(J), PRSE  38
              2COOR(J+1),SUBLEN(J)                                               PRSE  39
000247     152 FORMAT (X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,F8.1,X,F7.3,X,    PRSE  40
              1F8.1,X,F7.3,X,F8.1,X,F7.3,X,F8.1,X,F8.1,X,F7.2)                   PRSE  41
000247         GO TO 155                                                         PRSE  42
         C           PRINTS THE TABLE FOR A NON-COMPOSITE DESIGN                 PRSE  43
000253    1155 WRITE (6,1152)J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(J,2),         PRSE  44
              1PLATH(J,2),ISTEFL(J),CS(J),COOR(J),COOR(J+1),SUBLEN(J)            PRSE  45
000360    1152 FORMAT (X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,F8.1,X,F7.3,X,    PRSE  46
              1     F8.1,X,F8.1,X,F7.2)                                          PRSE  47
000360     155 CONTINUE                                                          PRSE  48
000366         IF (JPT3 .GT. 9) WRITE (6,1003)                                   PRSE  49
000400    1003 FORMAT (1H0,10X,28HTHE DESIGN IS NON-COMPOSITE.)                  PRSE  50
000400         RETURN                                                            PRSE  51
000401         END                                                               PRSE  52

SUBPROGRAM LENGTH
000550

UNUSED COMPILER SPACE
041100


               SUBROUTINE PRSTIF (NS,BEAR)                                       PRST   1
         C           PRINTS THE STIFFENER DESIGN INFORMATIO                      PRST   2
000005         DIMENSION BEAR(5,3)                                               PRST   3
         C           PRINT THE HEADING                                           PRST   4
000005         WRITE (6,201)                                                     PRST   5
000010     201 FORMAT (1H0,30HBEARING STIFFENER DESIGNS...../)                   PRST   6
000010         [illegible] = 0                                                   PRST   7
         C           PRINT THE INFORMATION FOR EACH SUPPORT                      PRST   8
```

```
000011          DO 200 I = 1, NS                                                PRST    9
000014          IF (ABS(BEAR(I,1)) .LE. .00001) GO TO 200                       PRST   10
000020          KPT = 1                                                         PRST   11
000021          WRITE (6,300) I, BEAR(I,1), BEAR(I,2), BEAR(I,3)                PRST   12
000043      200 CONTINUE                                                        PRST   13
000047      300 FORMAT (10X,12HSUPPORT NO. ,I1,5X,9HUSE 2 - ,F7.3,2H X,F6.3,   PRST   14
               14H PLATES. CONNECT ONE TO EACH SIDE WITH A ,F7.4,19H INCH FILLET PRST   15
               2WELD.)                                                          PRST   16
          C         ONLY PRINT IF NO STIFFENERS ARE REQUIRED                    PRST   17
000047          IF (KPT .EQ. 0) WRITE (6,202)                                   PRST   18
000055      202 FORMAT (22X,36HBEARING STIFFENERS ARE NOT REQUIRED.)            PRST   19
000055          RETURN                                                          PRST   20
000056          END                                                             PRST   21

SUBPROGRAM LENGTH
000125


UNUSED COMPILER SPACE
042700



                SUBROUTINE PRSTR (STRESS,NA)                                    PRSR    1
          C         PRINTS THE ENTIRE STRESS TABLE FOR A GIRDER                 PRSR    2
000005          COMMON/ONE/JPT3,IPT10                                           PRSR    3
000005          DIMENSION STRESS(81,4,2)                                        PRSR    4
          C         PRINT THE HEADING FOR THE TABLE                             PRSR    5
000005          WRITE (6,890)                                                   PRSR    6
000010      890 FORMAT (1H1,37HSTRESSES ALONG THE DESIGN GIRDER.....//5X,2HPT,  PRSR    7
               15X,10HBOT STRESS,5X,10HTOP STRESS,4X,11HCONC STRESS,4X,         PRSR    8
               212HSHEAR STRESS,11X,22HSAFE STRESS INDICATORS//)                PRSR    9
          C         PRINT THE STRESSES FOR EACH POINT                           PRSR   10
000010          DO 2500 I = 1, NA                                               PRSR   11
000013     2500 WRITE (6,891) I, STRESS(I,1,1), STRESS(I,2,1), STRESS(I,3,1),   PRSR   12
               1STRESS(I,4,1), STRESS(I,1,2), STRESS(I,2,2), STRESS(I,3,2),     PRSR   13
               2STRESS(I,4,2)                                                   PRSR   14
000064      891 FORMAT (4X,I3,3X,4(F10.3,5X),8X,4(F5.1,X))                      PRSR   15
          C         PRINT THE NON-COMPOSITE DESIGN STATEMENT                    PRSR   16
000064          IF (JPT3 .GT. 9) WRITE (6,870)                                  PRSR   17
000074      870 FORMAT (1H0,10X,28HTHE DESIGN IS NON-COMPOSITE.)                PRSR   18
000074          RETURN                                                          PRSR   19
000075          END                                                             PRSR   20

SUBPROGRAM LENGTH
000142

UNUSED COMPILER SPACE
042600


                SUBROUTINE BEAC                                                 BEAC    1
```

```
          C         SOLVES FOR THE REACTION INFLUENCE LINES                     REAC    2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), REAC    3
               1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),      REAC    4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,   REAC    5
               3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),EM,SPAN(4),IOSP(5), REAC    6
               4TMOM1(80),NA,LENGTH,RO(81),SUBLEN(80),NS                        REAC    7
000002          COMMON/ONE/JPI3,IPTIO                                           REAC    8
000002          DIMENSION RI1(3), RI2(3), RI3(3),RT(5,81)                       REAC    9
000002          REAL IX, IY, ISTEEL, ILCON, IHCON, MR, LENGTH                   REAC   10
          C         ZERO OUT VARIOUS MATRICES                                   REAC   11
000002          DO 31 I = 1, 5                                                  REAC   12
000004          DO 30 J = 1, 81                                                 REAC   13
000005          DO 30 K = 1, 3                                                  REAC   14
000006       30 RIL(I,J,K) = 0.0                                                REAC   15
000017          DO 31 II = 1, 5                                                 REAC   16
000020       31 CC(I,II) = 0.0                                                  REAC   17
000027          DO 33 I = 1, 80                                                 REAC   18
000030       33 TMOM1(I) = 0.0                                                  REAC   19
000033          DO 34 I = 1, 81                                                 REAC   20
000034       34 RO(I) = 0.0                                                     REAC   21
          C         BEGIN TO CALCULATE FOR EACH ANALYSIS POINT                  REAC   22
000037          DO 39 I = 1, NA                                                 REAC   23
          C         CALCULATE REACTIONS AND MOMENTS FOR A SIMPLE BEAM           REAC   24
000040          RR = COOR(I)/LENGTH                                             REAC   25
000042          RL = 1.0 - RR                                                   REAC   26
000044          DO 32 J = 1, I                                                  REAC   27
000046       32 RO(J) = RL * COOR(J)                                            REAC   28
000052          MM = I + 1                                                      REAC   29
000054          DO 35 J = MM, NA                                                REAC   30
          C         CALL ON EQSET TO SET UP THE RIL ARRAY WHICH IS THE RIGHT    REAC   32
          C         HAND SIDE OF THE EQUATION.                                  REAC   33
000055       35 RO(J) = RL * COOR(J) - (COOR(J) - COOR(I))                      REAC   31
000064          CALL EQSET(RO,SUBLEN,LENGTH,COOR,NF,ISTEEL,EM,NS,IOSP,SPAN,RI1) REAC   34
000076          CALL EQSET(RO,SUBLEN,LENGTH,COOR,NF,ILCON,EM,NS,IOSP,SPAN,RI2)  REAC   35
000111          CALL EQSET(RO,SUBLEN,LENGTH,COOR,NF,IHCON,EM,NS,IOSP,SPAN,RI3)  REAC   36
000124     1335 K = 3                                                           REAC   37
000125          DO 44 J = 1, 3                                                  REAC   38
000127          RIL(1,I,J) = 1.0                                                REAC   39
000134          RIL(2,I,J) = COOR(I)                                            REAC   40
000140          IF (K .GT. NS) GO TO 44                                         REAC   41
000144          RIL(K,I,1) = RI1(J)                                             REAC   42
000147          RIL(K,I,2) = RI2(J)                                             REAC   43
000151          RIL(K,I,3) = RI3(J)                                             REAC   44
000154          K = K + 1                                                       REAC   45
000155       44 CONTINUE                                                        REAC   46
000157       39 CONTINUE                                                        REAC   47
          C         ASSEMBLE COEFFICIENT MATRIX AND SOLVE THE EQUATIONS         REAC   48
000162          MN = NS - 1                                                     REAC   49
000164          DO 49 K = 1, 3                                                  REAC   50
000165          DO 40 I = 1,NS                                                  REAC   51
000166          CC(1,I) = 1.0                                                   REAC   52
000171          IF (I .NE. 1) GO TO 41                                          REAC   53
000173          CC(2,1) = 0.0                                                   REAC   54
000174          GO TO 40                                                        REAC   55
000174       41 CC(2,I) = CC(2,I-1) + SPAN(I-1)                                 REAC   56
000202       40 CONTINUE                                                        REAC   57
000205          DO 45 I = 2, MN                                                 REAC   58
000206          DO 45 J = 1, NS                                                 REAC   59
```

```
000207           JJ = IOSP(J)                                                  REAC  60
000211        45 CC(I+1,J) = RIL(I+1,JJ,K)                                     REAC  61
000224           DO 47 I = 1, NS                                               REAC  62
000226           DO 47 J = 1, NA                                               REAC  63
000227        47 RI(I,J) = RIL(I,J,K)                                          REAC  64
000244           CALL SOLVE (NS,NA,CC,RI,NA)                                   REAC  65
000247           DO 48 I = 1, NS                                               REAC  66
000251           DO 48 J = 1, NA                                               REAC  67
000252        48 RIL(I,J,K) = RI(I,J)                                          REAC  68
000267        49 CONTINUE                                                      REAC  69
000271      1400 RETURN                                                        REAC  70
000272           END                                                           REAC  71

SUBPROGRAM LENGTH
001162

UNUSED COMPILER SPACE
041500


                 SUBROUTINE READIN                                             READ   1
          C         READS IN THE DATA FOR THE PROBLEM                          READ   2
000002           COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), READ   3
                1BAR(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),    READ   4
                2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF, READ   5
                3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),F ,SPAN(4),IOSP(5), READ   6
                4TIDEI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLO(10),PROP(3),READ   7
                5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),DEAR(5,3),READ   8
                6COST(9)                                                       READ   9
000002           COMMON/ONE/JPT3,IPT10                                         READ  10
000002           COMMON/TWO/LPCM                                               READ  11
000002           COMMON/THREE/AS,NCS,IPT6                                      READ  12
000002           COMMON/TEN/HAUNCH                                             READ  13
000002           COMMON/D/IDEF(20),DESDEF(20,4),NDEF,PERCEN,ICYC               READ  14
000002           COMMON/F/IPT2                                                 READ  15
000002           COMMON/F/IPT7                                                 READ  16
000002           REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                       READ  17
000002           DIMENSION TITLE(36), EPS(4), LPCM(10)                         READ  18
000002           INTEGER EPS,TITLE                                             READ  19
          C         ZERO THE REQUIRED INPUT ARRAYS                             READ  20
000002           DO 510 I = 1, 80                                              READ  21
000004           NOS(I) = 0                                                    READ  22
000005           ISTEEL(I) = 0.0                                               READ  23
000006           CS(I) = 0.0                                                   READ  24
000007           ILCON(I) = 0.0                                                READ  25
000010           CL(I) = 0.0                                                   READ  26
000011           IHCON(I) = 0.0                                                READ  27
000012           CH(I) = 0.0                                                   READ  28
000013           SUBLEN(I) = 0.0                                               READ  29
000014           DO 510 K = 1, 2                                               READ  30
000016           PLATH(I,K) = 0.0                                              READ  31
000021       510 PLWID(I,K) = 0.0                                              READ  32
000030           SLABWD = 0.0                                                  READ  33
000031           SLABTH = 0.0                                                  READ  34
```

```
000032          MR = 0.0                                                       READ  35
000033          DO 501 I = 1, 4                                                READ  36
000034          SPAN(I) = 0.0                                                  READ  37
000035      501 EPS(I) = 0                                                     READ  38
000040          DO 502 I = 1, 36                                               READ  39
000041      502 TITLE(I) = 0                                                   READ  40
000044          DO 503 I = 1, 5                                                READ  41
000045      503 IOSP(I) = 0                                                    READ  42
000050          DO 504 I = 1, 10                                               READ  43
000051          LPCM(I) = 0                                                    READ  44
000052      504 TLOD(I) = 0.0                                                  READ  45
000055          DO 505 I = 1, 9                                                READ  46
000056      505 UNLD(I) = 0.0                                                  READ  47
000061          DO 506 I = 1, 3                                                READ  48
000062      506 PROP(I) = 0.0                                                  READ  49
000065          DO 508 I = 1, 20                                               READ  50
000066      508 IDEF(I) = 0                                                    READ  51
000071          DO 509 I = 1, 61                                               READ  52
000072      509 COOR(I) = 0.0                                                  READ  53
        C            READ TITLE AND INPUT PARAMETER DATA CARDS                 READ  54
000075          READ (5,100) (TITLE(I), I = 1, 36)                             READ  55
000106      100 FORMAT (18A4/18A4)                                             READ  56
000106          READ (5,102) IPT1,IPT2,IPT3,IPT4,IPT5,IPT6,IPT7,IPT8,IPT9,IPT10   READ  57
000136      102 FORMAT (10I5)                                                  READ  58
        C            PRINT TITLE OF THE PROBLEM                                READ  59
000136          WRITE (6,101) (TITLE(I), I = 1, 36)                            READ  60
000150      101 FORMAT (1H1///26X,18A4///26X,18A4////)                         READ  61
        C            PRINT THE HEADING IF IPT10 IS GREATER THAN 2              READ  62
000150          IF (IPT10 .LT. 3) GO TO 1010                                   READ  63
000153          WRITE (6,103)                                                  READ  64
000156      103 FORMAT (1H0,30X,66HTHIS PROBLEM WILL BE SOLVED USING THE TENTH EDIREAD  65
               1TION OF THE HIGHWAY/25X,70HBRIDGE SPECIFICATIONS PUBLISHED IN 1969READ  66
               2 BY THE AMERICAN ASSOCIATION OF/25X,69HSTATE HIGHWAY OFFICIALS.  TREAD  67
               3HE OPTIMUM PROGRAM USED IN THIS DESIGN WAS/25X,66HDEVELOPED BY ROBREAD  68
               4ERT H. HUSEK DURING HIS STUDY FOR A MASTERS DEG EE/25X,69HAT PURDUREAD  69
               5E UNIVERSITY.  THE RESEARCH WAS FINANCED BY THE JOINT HIGHWAY/25X,READ  70
               670HRESEARCH BOARD.  THE RESEARCH WAS DIRECTED BY THE FOLLOWING MEMREAD  71
               7BERS OF/25X,25HTHE RESEARCH COMMITTEE.../50X,25 1.  PROFESSOR J. TREAD  72
               8. BAUNI/50X,21H2.  PROFESSOR A. D. M. LEWIS/50X,               READ  73
               920H3.  PROFESSOR C. W. LOVELL)                                 READ  74
000156          WRITE (6,104)                                                  READ  75
000162      104 FORMAT (1H0,24X,72HTHE PROGRAM, AS DEVELOPED, WILL DESIGN THE OPTIREAD  76
               1MUM HIGHWAY BRIDGE GIRDER/25X,67HCOMPOSED OF A ROLLED SECTION ACTIREAD  77
               2NG COMPOSITELY WITH THE DECK SLAB./25X,73HTHE PROGRAM CAN SOLVE A READ  78
               3FOUR SPAN CONTINUOUS BRIDGE GIRDER AS WELL AS THE/25X,71HSIMPLE SPREAD  79
               4AN GIRDER PROBLEM.  THE SOLUTION IS FOUND BY OPTIMIZING EITHER/   READ  80
               525X,64HTHE WEIGHT OR THE TOTAL COST INCLUDING THE COSTS OF FABRICAREAD  81
               6TION./////)                                                    READ  82
        C            PRINT THE INPUT PARAMETERS IF IPT10 IS GREATER THAN 1     READ  83
000162     1010 IF (IPT10 .LT. 2) GO TO 1011                                   READ  84
000165          WRITE (6,105) IPT1,IPT2,IPT3,IPT4,IPT5,IPT6,IPT7,IPT8,IPT9,IPT10   READ  85
000214      105 FORMAT (1H0,24X,28HTHE INPUT PARAMETERS ARE.../34X,8HIPT1 = ,I2,READ  86
               15X,27HSUPPLEMENT INFORMATION/34X,8HIPT2 = ,I2,5X,24HTYPE OF LOADIREAD  87
               2NG REQUIRED/34X,8HIPT3 = ,I2,5X,28HINITIAL GIRDER CROSS-SECTION / READ  88
               334X,8HIPT4 = ,I2,5X,13HUNITS OF DATA/34X,8HIPT5 = ,I2,5X,40HACTIONREAD  89
               4AL LOADINGS CONSIDERED IN DESIGNATION/34X,8HIPT6 = ,I2,5X,20HTYPE READ  90
               5OF WELD DETAIL/34X,8HIPT7 = ,I2,5X,13HSTRESS CYCLES/34X,       READ  91
               68HIPT8 = ,I2,5X,28HMATERIAL USED IN THE FLANGES/34X,8HIPT9 = ,  READ  92
```

```
             7I2,5X,19HDEFLECTIONS DESIRED/34X,8HIPT1 = ,I2,5X,16HOUTPUT PARAMEREAD  93
             1TERS)                                                            READ  94
000214         WRITE (6,106)                                                   READ  95
000220     106 FORMAT (1H0,24X,73HSEE INPUT INSTRUCTIONS FOR THE DEFINITIONS OF TREAD  96
             1HE INPUT PARAMETER VALUES.//)                                    READ  97
         C        READ THE LENGTHS OF SPANS                                    READ  98
000220    1011 READ (5,110) (SPAN(I), I = 1, 4)                                READ  99
000232     110 FORMAT (4F10.4)                                                 READ 100
         C        DETERMINE THE NUMBER OF SUPPORTS AND THE TOTAL LENGTH        READ 101
000232         NS = 1                                                          READ 102
000233         LENGTH = 0.0                                                    READ 103
000234         DO 111 I = 1, 4                                                 READ 104
000236         LENGTH = LENGTH + SPAN(I)                                       READ 105
000240         IF (SPAN(I) .LT. .001) GO TO 112                                READ 106
000243     111 NS = NS + 1                                                     READ 107
000246     112 MN = NS - 1                                                     READ 108
         C        DETERMINE THE NUMBER OF ELEMENTS PER SPAN                    READ 109
         C        DETERMINE THE COORDINATE ARRAY                               READ 110
         C        DETERMINE THE INDEX OF THE SUPPORT POINTS                    READ 111
         C        DETERMINE THE SUBLENGTH ARRAY                                READ 112
000250         IF (IPT1 .NE. 0) GO TO 113                                      READ 113
000251         DO 114 I = 1, MN                                                READ 114
000253     114 EPS(I) = 10                                                     READ 115
000257         GO TO 115                                                       READ 116
000257     113 IF (IPT1 .NE. 2) GO TO 120                                      READ 117
000261         READ (5,116) (EPS(I), I = 1, MN)                                READ 118
000274     116 FORMAT (4I10)                                                   READ 119
000274     115 IOSP(1) = 1                                                     READ 120
000275         NE = 0                                                          READ 121
000276         NA = 1                                                          READ 122
000277         COOR(1) = 0.0                                                   READ 123
000300         DO 117 I = 1, MN                                                READ 124
000302         IOSP(I+1) = IOSP(I) + EPS(I)                                    READ 125
000304         NE = NE + EPS(I)                                                READ 126
000306     117 NA = NA + EPS(I)                                                READ 127
000311         DO 118 I = 1, MN                                                READ 128
000313         IA = IOSP(I)                                                    READ 129
000315         IB = IOSP(I+1) - 1                                              READ 130
000317         SUB = SPAN(I)/EPS(I)                                            READ 131
000321         DO 119 J = IA, IB                                               READ 132
000323         SUBLEN(J) = SUB                                                 READ 133
000325     119 COOR(J+1) = COOR(J) + SUBLEN(J)                                 READ 134
000331     118 CONTINUE                                                        READ 135
000334         GO TO 125                                                       READ 136
000334     120 READ (5,121) NA                                                 READ 137
000342     121 FORMAT (I10)                                                    READ 138
000342         READ (5,122) (COOR(I), I = 1, NA)                               READ 139
000355     122 FORMAT (7F10.2)                                                 READ 140
000355         IOSP(1) = 1                                                     READ 141
000356         NE = NA - 1                                                     READ 142
000360         DO 123 I = 1, NE                                                READ 143
000362     123 SUBLEN(I) = COOR(I+1) - COOR(I)                                 READ 144
000367         J = 1                                                           READ 145
000370         DO 124 I = 1, MN                                                READ 146
000371         TEMP = 0.0                                                      READ 147
000372         K = 1                                                           READ 148
000373         TEMP = TEMP + SUBLEN(J)                                         READ 149
000376     127 IF (TEMP .GT. SPAN(I)) GO TO 125                                READ 150
```

```
000575          WRITE (6,1031) (NAME(I), I = 1, 18)                             READ 209
000607     1031 FORMAT (1H0,20X,6(A7,2X)/21X,6(A7,2X)/21X,6(A7,2X))             READ 210
000607          GO TO 140                                                       READ 211
000610      139 WRITE (6,1030)                                                  READ 212
000614          WRITE (6,1002)                                                  READ 213
000620     1002 FORMAT (1H0,6X,4HNAME,8X,5HSAREA,6X,5HDEPTH,6X,5HFLWID,7X,4HFLTH, READ 214
               16X,5HWEBTH,7X,3HDAF,10X,2HIX,13X,2HIY,6X,5HNSECT//)             READ 215
000620          DO 135 I = 1, 18                                                READ 216
000622      135 WRITE (6,1004) NAME(I),SAREA(I),DEPTH(I),FLWID(I),FLTH(I),      READ 217
               1WEBTH(I),DAF(I),IX(I),IY(I),NSECT(I)                            READ 218
000653     1004 FORMAT (9X,A7,2(5X,F6.2),3(5X,F6.3),5X,F5.2,2(5X,F10.2),5X,I2)  READ 219
          C          READ THE COMPOSITE SECTION DATA IF REQUIRED                READ 220
000653      140 IF (IPT3 .LT. 10) GO TO 141                                     READ 221
000656          JPT4 = IPT3                                                     READ 222
000657          IPT4 = IPT3 - 10                                                READ 223
000660          WRITE (6,1141)                                                  READ 224
000664     1141 FORMAT (1H0/1H ,20X,49HTHE GIRDER DESIGN WILL BE A NON-COMPOSITE DREAD 225
               1ESIGN.)                                                         READ 226
000664          GO TO 142                                                       READ 227
000665      141 READ (5,143) SLABWD,SLABTH, MR, FPC, AS, HAUNCH                 READ 228
000705      143 FORMAT (5F10.2,F10.4)                                           READ 229
000705          JPT4 = IPT3                                                     READ 230
          C          PRINT THE COMPOSITE SECTION DATA IF REQUIRED               READ 231
000707          WRITE (6,1040) SLABWD, SLABTH, MR, FPC, AS, HAUNCH              READ 232
000726     1040 FORMAT (1H0,10X,83HTHE DESIGN WILL BE BASED ON A COMPOSITE SECTIONREAD 233
               1 HAVING THE FOLLOWING PROPERTIES..../20X,26HEFFECTIVE WIDTH OF SLAREAD 234
               2B = ,F7.2,8H INCHES./20X,17HSLAB THICKNESS = ,F6.2,8H INCHES./20X,READ 235
               316HMODULUS RATIO = ,F5.1,1H./20X,4HUSE ,F6.0,14H PSI CONCRETE./    READ 236
               420X,29HAREA OF SLAB REINFORCEMENT = ,F6.2,8H SQ. IN./20X,          READ 237
               522HTHICKNESS OF HAUNCH = ,F8.4,8H INCHES.)                      READ 238
          C          DETERMINE THE INITIAL SECTION PROPERTIES                   READ 239
000726      142 IF (IPT4 .NE. 0) GO TO 149                                      READ 240
000727          DO 145 I = 1, NF                                                READ 241
000731          NOS(I) = 18                                                     READ 242
000733          DO 145 K = 1, 2                                                 READ 243
000734          PLWID(I,K) = 0.0                                                READ 244
000737      145 PLATH(I,K) = 0.0                                                READ 245
000747          GO TO 150                                                       READ 246
000747      149 IF (IPT4 .NE. 2) GO TO 144                                      READ 247
000751          DO 146 I = 1, NF                                                READ 248
000753      146 READ (5,147) NOS(I),PLWID(I,1),PLATH(I,1),PLWID(I,2),PLATH(I,2)  READ 249
000773      147 FORMAT (I10,4F10.4)                                             READ 250
000773          GO TO 150                                                       READ 251
000773      144 READ (5,147) NOS(1),PLWID(1,1),PLATH(1,1),PLWID(1,2),PLATH(1,2)  READ 252
001011          DO 148 I = 2, NF                                                READ 253
001013          NOS(I) = NOS(1)                                                 READ 254
001015          DO 148 K = 1, 2                                                 READ 255
001016          PLWID(I,K) = PLWID(1,K)                                         READ 256
001023      148 PLATH(I,K) = PLATH(1,K)                                         READ 257
001032      150 CALL SECTIC                                                     READ 258
          C          PRINT THE INITIAL PROPERTIES IF IPT10 IS GREATER THAN 1    READ 259
001033          IF (IPT10 .LT. 2) GO TO 160                                     READ 260
001036          WRITE (6,4700)                                                  READ 261
001041     4700 FORMAT (1H1///35X,54HPROPERTIES OF THE GIRDER USED FOR THE INITIALREAD 262
               1 ANALYSIS//)                                                    READ 263
001041          IF (JPT4 .GT. 9) GO TO 1160                                     READ 264
001045          WRITE (6,151)                                                   READ 265
001050      151 FORMAT (1H0,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HBOT,4X,3HBOT,      READ 266
```

```
                 155X,4HLEFT,4X,5HRIGHT/2X,1HI,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,
                 25HPLWID,2X,5HPLATH,2X,6HISTEEL,5X,2HCS,5X,5HILCON,5X,2HCL,5X,
                 35HIHCON,5X,2HCH,6X,4HCOOR,5X,4HCOOR,3X,6HSUBLEN//)
001050           GO TO 1170
001051      1160 WRITE (6,1151)
001055      1151 FORMAT (1H0,5X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HBOT,4X,3HBOT,
                 122X,4HLEFT,4X,5HRIGHT/2X,1HI,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,
                 25HPLWID,2X,5HPLATH,2X,6HISTEEL,5X,2HCS,6X,
                 3                        4HCOOR,5X,4HCOOR,3X,6HSUBLEN//)
001055      1170 DO 155 J = 1, EE
001057           K = POS(J)
001061           IF (IPT3 .GT. 4) GO TO 1155
001064           WRITE (6,152) J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(J,2),
                 1PLATH(J,2),ISTEEL(J),CS(J),ILCON(J),CL(J),IHCON(J),CH(J),COOR(J),
                 2COOR(J+1),SUBLEN(J))
001125       152 FORMAT (X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,F8.1,X,F7.3,X,
                 1F8.1,X,F7.3,X,F8.1,X,F7.3,X,F8.1,X,F8.1,X,F7.2)
001125           GO TO 155
001126      1155 WRITE (6,1152)J,NAME(K),PLWID(J,1),PLATH(J,1),PLWID(J,2),
                 1PLATH(J,2),ISTEEL(J),CS(J),COOR(J),COOR(J+1),SUBLEN(J)
001160      1152 FORMAT (X,I2,X,A7,X,F6.2,X,F6.3,X,F6.2,X,F6.3,X,F8.1,X,F7.3,X,
                 1        F8.1,X,F8.1,X,F7.2)
001160       155 CONTINUE
001163           WRITE (6,1156)
001166      1156 FORMAT (1H0,40HTHE UNITS OF THE ABOVE TABLE ARE INCHES.)
            C           READ IN THE LOAD INFORMATION
001166       160 IF (IPT5 .NE. 0) GO TO 170
001167           READ (5,161) LDIS, PLL, SLABWT, SDL, GDI, SLL, OI, OIL
001213       161 FORMAT (A4,F6.3,6F10.4)
001213           IF (LDIS .EQ. 4H H10) Q = PLL * .5
001217           IF (LDIS .EQ. 4H H15 .OR. LDIS .EQ. 4HHS15) Q = PLL * .75
001230           IF (LDIS .EQ. 4H H20 .OR. LDIS .EQ. 4HHS20) Q = PLL
001240           IF (IPT2 .EQ. 2) GO TO 175
001242           UNLD(1) = Q * .0533
001244           UNLD(2) = Q * 18.
001246           UNLD(3) = Q * 26.
001250           UNLD(4) = SDL
001251           UNLD(5) = GDI
001253           UNLD(6) = (1.0 - OIL) * PLL * 24.
001256           UNLD(7) = SLL
001260           UNLD(8) = OI
001261           UNLD(9) = SLABWT
001263       175 IF (IPT2 .EQ. 1) GO TO 176
001265           TRLD(1) = Q * 8.
001267           TRLD(2) = Q * 32.
001271           TRLD(4) = 168.
001272           TRLD(7) = SDL
001274           TRLD(8) = GDI
001275           IF (LDIS .EQ. 4HHS15 .OR. LDIS .EQ. 4HHS20) GO TO 177
001305           TRLD(3) = 0.0
001306           TRLD(5) = 90000.
001307           TRLD(6) = 90001.
001311           GO TO 178
001311       177 TRLD(3) = Q * 32.
001313           TRLD(5) = 168.
001315           TRLD(6) = 360.
001316       178 TRLD(9) = OI
001320           TRLD(10) = SLABWT
```

```
001321          GO TO 176                                                         READ 325
001322      170 IF (IPT2 .EQ. 1) GO TO 179                                        READ 326
001324          READ (5,180) (TRLO(I), I = 1, 10)                                 READ 327
001336      180 FORMAT (10F7.3)                                                   READ 328
001336      179 IF (IPT2 .EQ. 2) GO TO 176                                        READ 329
001340          READ (5,181) (UNLO(I), I = 1, 9)                                  READ 330
001352      181 FORMAT (9F8.4)                                                    READ 331
          C         CONVERT THE LOAD INFORMATION TO UNITS OF INCHES               READ 332
001352      176 IF (IPT4 .EQ. 0) GO TO 190                                        READ 333
001353          IF (IPT5 .EQ. 0) GO TO 185                                        READ 334
001354          TRLO(4) = TRLO(4) * 12.                                           READ 335
001356          TRLO(5) = TRLO(5) * 12.                                           READ 336
001357          TRLO(6) = TRLO(6) * 12.                                           READ 337
001360          UNLO(1) = UNLO(1)/12.                                             READ 338
001361      185 TRLO(7) = TRLO(7)/12.                                             READ 339
001363          TRLO(8) = TRLO(8)/12.                                             READ 340
001364          TRLO(10) = TRLO(10)/12.                                           READ 341
001366          UNLO(4) = UNLO(4)/12.                                             READ 342
001367          UNLO(5) = UNLO(5)/12.                                             READ 343
001371          UNLO(7) = UNLO(7)/12.                                             READ 344
001372          UNLO(9) = UNLO(9)/12.                                             READ 345
          C         PRINT THE LOAD INFORMATION                                    READ 346
001374      190 CONTINUE                                                          READ 347
001374          WRITE (6,195)                                                     READ 348
001400      195 FORMAT (1H1,30X,24HLOADING INFORMATION.....//)                    READ 349
001400          IF (IPT5 .EQ. 0) WRITE (6,200) LDIS                               READ 350
001407      200 FORMAT (1H0,30X,25HTHE DESIGN IS BASED ON A ,A4, 8H LOADING///)   READ 351
001407          IF (IPT2 .EQ. 1) GO TO 191                                        READ 352
001411          WRITE (6,201)                                                     READ 353
001415      201 FORMAT (40X,25HTRUCK LOAD INFORMATION...//)                       READ 354
001415          WRITE (6,202) TRLO(1),TRLO(2)                                     READ 355
001425      202 FORMAT (50X,18HFIRST AXLE LOAD = ,26X,F8.3,5H KIPS/50X,19HSECOND AREAD 356
               1XLE LOAD = ,25X,F8.3,5H KIPS)                                     READ 357
001425          TP = TRLO(4)/12.                                                  READ 358
001427          WRITE (6,203) TP                                                  READ 359
001435      203 FORMAT (50X,40HSPACING BETWEEN FIRST AND SECOND AXLE = ,4X,F8.3,  READ 360
               15H FEET)                                                          READ 361
001435          IF (IPT5.EQ.1.OR.LDIS.EQ.4HHS15.OR.LDIS.EQ.4HHS20) GO TO 192      READ 362
001450          GO TO 193                                                         READ 363
001451      192 TP1 = TRLO(5)/12.                                                 READ 364
001453          TP2 = TRLO(6)/12.                                                 READ 365
001455          WRITE (6,204) TRLO(3), TP1, TP2                                   READ 366
001467      204 FORMAT (50X,18HTHIRD AXLE LOAD = ,26X,F8.3,5H KIPS/50X,44HMINIMUM READ 367
               1SPACING OF SECOND AND THIRD AXLES = ,F8.3,5H FEET/50X,44HMAXIMUM SREAD 368
               2PACING OF SECOND AND THIRD AXLES = ,F8.3,5H FEET)                 READ 369
001467      193 TP1 = TRLO(7) * 12.                                               READ 370
001471          TP2 = TRLO(8) * 12.                                               READ 371
001473          TP3 = TRLO(10) * 12.                                              READ 372
001474          WRITE (6,205) TP1, TP2, TP3                                       READ 373
001506      205 FORMAT (50X,25HSUPERIMPOSED DEAD LOAD = ,19X,F8.3,8H KIPS/FT/50X, READ 374
               121HGIRDER DEAD WEIGHT = ,23X,F8.3,8H KIPS/FT/50X,                 READ 375
               222HDEAD WEIGHT OF SLAB = ,22X,F8.3,8H KIPS/FT)                    READ 376
001506          IF (TRLO(9) .GT. .1) WRITE (6,206)                                READ 377
001514      206 FORMAT (51X,29HNO IMPACT FACTOR WILL BE USED///)                  READ 378
001514      191 IF (IPT2 .EQ. 2) GO TO 249                                        READ 379
001516          WRITE (6,210)                                                     READ 380
001522      210 FORMAT (1H0,39X,27HLANE LOADING INFORMATION...//)                 READ 381
001522          TP = UNLO(1) * 12.                                                READ 382
```

```
001524          WRITE (6,211) TP, UNLO(2), UNLO(3)                                      READ 383
001536      211 FORMAT (50X,20HLIVE LANE LOADING = ,24X,F8.5,8H KIPS/FT/50X,            READ 384
               131HCONCENTRATED LOAD FOR MOMENT = ,13X,F8.3,5H KIPS/50X,                READ 385
               230HCONCENTRATED LOAD FOR SHEAR = ,14X,F8.3,5H KIPS)                     READ 386
001536          TP1 = UNLO(4) * 12.                                                     READ 387
001540          TP2 = UNLO(5) * 12.                                                     READ 388
001542          TP3 = UNLO(9) * 12.                                                     READ 389
001543          WRITE (6,205) TP1, TP2, TP3                                             READ 390
001555          TP1 = UNLO(7) * 12.                                                     READ 391
001557          WRITE (6,212) TP1, UNLO(6)                                              READ 392
001567      212 FORMAT (50X,21HSIDEWALK LIVE LOAD = ,23X,F8.3,8H KIPS/FT/50X,           READ 393
               134HCONCENTRATED INTERSTATE LOADING = ,10X,F8.3,5H KIPS)                 READ 394
001567          IF (UNLO(8) .GT. .1) WRITE (6,206)                                      READ 395
001575      249 IF (TRLO(9) .LT. .1 .AND. UNLO(8) .LT. .1) WRITE (6,215)                READ 396
001610      215 FORMAT (1H0,39X,34HIMPACT IS INCLUDED IN THE ANALYSIS)                  READ 397
          C         READ THE MATERIAL PROPERTIES IF REQUIRED                            READ 398
001610      250 IF (IPT8 .EQ. 1) GO TO 260                                              READ 399
001612          PROP(1) = 20.0                                                          READ 400
001614          PROP(2) = 12.0                                                          READ 401
001615          PROP(3) = 36.0                                                          READ 402
001617          EM = 29000.                                                             READ 403
001620          FC = .4 * FPC/1000.                                                     READ 404
001623          GO TO 270                                                               READ 405
001623      260 READ (5,261) PROP(3)                                                    READ 406
001631      261 FORMAT (F10.2)                                                          READ 407
001631          PROP(1) = .55 * PROP(3)                                                 READ 408
001633          PROP(2) = .33 * PROP(3)                                                 READ 409
001635          EM = 29000.                                                             READ 410
001636          FC = .4 * FPC/1000.                                                     READ 411
          C         PRINT THE MATERIAL PROPERTIES                                       READ 412
001641      270 CONTINUE                                                                READ 413
001641          WRITE (6,275) PROP(1), PROP(2), PROP(3), EM, FC                         READ 414
001657      275 FORMAT (1H0,39X,22HMATERIAL PROPERTIES...//50X,27HALLOWABLE BENDINREAD 415
               1G STRESS = ,17X,F8.3,4H KSI/50X,25HALLOWABLE SHEAR STRESS = ,19X  READ 416
               2F8.3,4H KSI/50X,15HYIELD STRESS = ,29X,F8.3,4H KSI/50X,                 READ 417
               324HMODULUS OF ELASTICITY = ,18X,F10.3,4H KSI/50X,                       READ 418
               428HALLOWABLE CONCRETE STRESS = ,16X,F8.3,4H KSI)                        READ 419
001657      300 CONTINUE                                                                READ 420
          C         READ THE COST INFORMATION IF REQUIRED                               READ 421
001657          IF (IPT6 .GT. 0) GO TO 1300                                             READ 422
001662          READ(5,1301) (COST(I), I = 1, 9)                                        READ 423
001673     1301 FORMAT (9F6.2)                                                          READ 424
          C         PRINT THE COST INFORMATION                                          READ 425
001673          WRITE (6,1302) COST(1),COST(2),COST(3),COST(4),COST(5),COST(6),         READ 426
               1COST(7),COST(8)                                                         READ 427
001717     1302 FORMAT (1H0,39X,24HUNIT COST INFORMATION...//50X,                       READ 428
               121HCOST OF WF SECTION = ,23X,F8.3,9H DOLS/LBS/50X,13HPLATE COST = ,READ 429
               231X,F8.3,9H DOLS/LBS/50X,                     29HCOST OF BEARING STIFFEREAD 430
               3NERS = ,15X,F8.3,9H DOLS/LBS/50X,27HCOST OF SHEAR CONNECTORS = ,  READ 431
               417X,F8.3,9H DOLS/LBS/50X,33HCOST OF CONNECTOR INSTALLATION = ,11X,READ 432
               5F8.3,15H DOLS/CONNECTOR/50X,18HFIXED WELD COST = ,26X,F8.3,       READ 433
               610H DOLS/WELD/50X,24HCOST OF WELD MATERIAL = ,20X,F8.3,           READ 434
               719H DOLS/CU IN OF WELD/50X,22HCOST OF WELD LENGTH = ,22X,F8.3,    READ 435
               818H DOLS/IN OF LENGTH)                                                  READ 436
001717     1300 CONTINUE                                                                READ 437
          C         DETERMINE THE NUMBER OF CYCLE INFORMATION AND PRINT                 READ 438
001717          IF (IPT7 .EQ. 0) GO TO 2300                                             READ 439
001720          IF (IPT7 .EQ. 2) GO TO 2301                                             READ 440
```

```
001722          READ(5,2302) NCYC, PERCEN                                               READ 441
001731     2302 FORMAT (I10,F10.2)                                                      READ 442
001731          WRITE (6,2303) PERCEN,NCYC                                              READ 443
001741     2303 FORMAT (1H1,20X,52HTHE DESIGN CYCLE IS REPEATED UNTIL THE FINAL MOREAD 444
               1MENTS/21X,22HARE CHANGED LESS THAN ,F6.2,22H PERCENT.  THE MAXIMUMREAD 445
               2/21X,20HNUMBER OF CYCLES IS ,I2,1H.////)                                READ 446
001741          GO TO 2350                                                              READ 447
001742     2301 READ (5,2304) NCYC                                                      READ 448
001750     2304 FORMAT (I10)                                                            READ 449
001750          PERCEN = 0.0                                                            READ 450
001751          WRITE (6,2305) NCYC                                                     READ 451
001757     2305 FORMAT (1H1,20X,40HTHE DESIGN CYCLE IS REPEATED A TOTAL OF ,I2,        READ 452
               17H TIMES.////)                                                          READ 453
001757          GO TO 2350                                                              READ 454
001760     2300 NCYC = 1                                                                READ 455
001761          PERCEN = 0.0                                                            READ 456
001762          WRITE (6,2306)                                                          READ 457
001766     2306 FORMAT (1H1,20X,50HTHE DESIGN CYCLE IS REPEATED ONLY ONCE.  THE BAREAD 458
               1SIC/21X,52HSECTION DETERMINED IN THE DESIGN CYCLE IS REDESIGNED/  READ 459
               221X,48HUSING THE NEW DESIGN LOADINGS AND THE PROGRAM IS/              READ 460
               321X,16HTHEN TERMINATED.////)                                            READ 461
          C         DETERMINE THE REQUIRED DEFLECTION LOCATIONS                         READ 462
001766     2350 IF (IPT9 .NE. 0) GO TO 2400                                             READ 463
001767          DO 2401 I = 1, MN                                                       READ 464
001771          ID = IOSP(I)                                                            READ 465
001773          DT = COOR(ID) + SPAN(I) / 2.                                            READ 466
001776          AS = 999999.                                                            READ 467
002000          DO 2402 J = 1, NA                                                       READ 468
002001          IF (ABS(COOR(J) - DT) .GT. AS) GO TO 2402                               READ 469
002007          IA = J                                                                  READ 470
002007          AS = ABS(COOR(J) - DT)                                                  READ 471
002011     2402 CONTINUE                                                                READ 472
002014          IDEF(I) = IA                                                            READ 473
002016     2401 CONTINUE                                                                READ 474
002020          NDEF = MN                                                               READ 475
002021          GO TO 2450                                                              READ 476
002021     2400 IF (IPT9 .NE. 1) GO TO 2460                                             READ 477
002023          IB = 1                                                                  READ 478
002024          DO 2410 I = 1, MN                                                       READ 479
002026          D = .2                                                                  READ 480
002030          ID = IOSP(I)                                                            READ 481
002031     2425 DT = COOR(ID) + D * SPAN(I)                                             READ 482
002035          AS = 999999.                                                            READ 483
002037          DO 2411 J = 1, NA                                                       READ 484
002040          IF (ABS(COOR(J) - DT) .GT. AS) GO TO 2411                               READ 485
002046          IA = J                                                                  READ 486
002046          AS = ABS(COOR(J) - DT)                                                  READ 487
002050     2411 CONTINUE                                                                READ 488
002053          IDEF(IB) = IA                                                           READ 489
002055          IB = IB + 1                                                             READ 490
002056          IF (D .LT. .4) GO TO 2421                                               READ 491
002061          IF (D .LT. .6) GO TO 2422                                               READ 492
002063          GO TO 2410                                                              READ 493
002064     2421 D = .5                                                                  READ 494
002066          GO TO 2425                                                              READ 495
002066     2422 D = .8                                                                  READ 496
002070          GO TO 2425                                                              READ 497
002070     2410 CONTINUE                                                                READ 498
```

```
002073          NDEF = IB - 1                                                   READ 499
002075          GO TO 2450                                                      READ 500
002075     2460 READ (5,2465) NDEF, (IDEF(I), I = 1, NDEF)                      READ 501
002112     2465 FORMAT (I10, 20I3)                                              READ 502
         C         PRINT THE REQUIRED DEFLECTION LOCATIONS                      READ 503
002112     2450 WRITE (6,2455) (IDEF(I), I = 1, NDEF)                           READ 504
002125     2455 FORMAT (1H0,20X,50HDEFLECTIONS ARE FOUND AT THE FOLLOWING POINTS.,READ 505
               1...//24X,20I3)                                                  READ 506
002125          RETURN                                                          READ 507
002126          END                                                             READ 508

SUBPROGRAM LENGTH
003672


UNUSED COMPILER SPACE
031000


                SUBROUTINE SFMIC                                                SFMC   1
         C         SFMIC SETS THE CONDITIONS FOR USING ICALC TO SOLVE FOR I VALUESSFMC   2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), SFMC   3
               1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),      SFMC   4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,   SFMC   5
               3SLABWD,SLABTH,MR                                                SFMC   6
000002          COMMON/ONE/JPT3,IPT10                                           SFMC   7
000002          COMMON/TWO/LPCM                                                 SFMC   8
000002          COMMON/TEN/HAUNCH                                               SFMC   9
000002          DIMENSION LPCM(10)                                              SFMC  10
000002          REAL IX, IY, ISTEEL, ILCON, IHCON, MR                           SFMC  11
000002          INTEGER NAME                                                    SFMC  12
         C         SET INITIAL ZERO VALUES                                      SFMC  13
000002          ISA = 0.                                                        SFMC  14
000003          H = 0.                                                          SFMC  15
000004          WDDIF = 0.                                                      SFMC  16
000005          THDIF = 0.                                                      SFMC  17
000006          IF (JPT3 .LT. 9) ISWID1 = SLABWD / (3. * MR)                    SFMC  18
000013          IF (JPT3 .LT. 9) ISWID2 = SLABWD / MR                           SFMC  19
         C         START ITERATION LOOP FOR EACH SUBELEMENT                     SFMC  20
000017          DO 25 I = 1, NE                                                 SFMC  21
         C         CHECK PREVIOUS SECTION FOR SIMILARITY TO THE CURRENT ONE     SFMC  23
000021          IF (I .EQ. 1) GO TO 22                                          SFMC  22
000023          K = I - 1                                                       SFMC  24
         C         CHECK FOR SIMILAR STEEL SECTIONS                             SFMC  25
000024          IF (NOS(I) .NE. NOS(K)) GO TO 22                                SFMC  26
         C         CHECK FOR SIMILAR TOP AND BOTTOM COVER PLATES                SFMC  27
000026          DO 26 KK = 1, 2                                                 SFMC  28
000027          THDIF = ABS(PLATH(I,KK) - PLATH(K,KK))                          SFMC  29
000036          IF (THDIF .GT. .00001) GO TO 22                                 SFMC  30
000042          WDDIF = ABS(PLWID(I,KK) - PLWID(K,KK))                          SFMC  31
000047          IF (WDDIF .GT. .00001) GO TO 22                                 SFMC  32
000052       26 CONTINUE                                                        SFMC  33
         C         SETS I VALUES TO THOSE OF THE SIMILAR SECTION                SFMC  34
000054          ISTEEL(I) = ISTEEL(K)                                           SFMC  35
000056          CS(I) = CS(K)                                                   SFMC  36
```

```
000060          ILCON(I) = ILCON(K)                                                SFMC  37
000061          CL(I) = CL(K)                                                      SFMC  38
000063          IHCON(I) = IHCON(K)                                                SFMC  39
000064          CH(I) = CH(K)                                                      SFMC  40
000066          GO TO 25                                                           SFMC  41
000066       22 CONTINUE                                                           SFMC  42
          C          CALCULATES I VALUES WHEN THERE IS NO SIMILAR SECTION          SFMC  43
000066       20 J = NOS(I)                                                         SFMC  44
000070          HP = DEPTH(J)/2. + PLATH(I,2)                                      SFMC  45
000073          CALL ICALC(PLWID(I,1),PLATH(I,1),PLWID(I,2),PLATH(I,2),DEPTH(J),   SFMC  46
               1SAREA(J),IA(J),HP,0.,CS(I),ISTEEL(I))                              SFMC  47
000114          D = 0.                                                             SFMC  48
000115          IF (JPT3 .GE. 10) GO TO 225                                        SFMC  49
000120          C = 0.                                                             SFMC  50
000121          TSA = SAREA(J)+PLWID(I,1)*PLATH(I,1)+PLWID(I,2)*PLATH(I,2)         SFMC  51
000127          H = 2. * (DEPTH(J) + PLATH(I,2) - CS(I))                           SFMC  52
000134          CALL ICALC(ISWID1,SLABTH,D,C,H,TSA,ISTEEL(I),CS(I),HAUNCH,CL(I),   SFMC  53
               1ILCON(I))                                                          SFMC  54
000156          CALL ICALC(ISWID2,SLABTH,D,C,H,TSA,ISTEEL(I),CS(I),HAUNCH,CH(I),   SFMC  55
               1IHCON(I))                                                          SFMC  56
000201          GO TO 25                                                           SFMC  57
000202      225 ILCON(I) = ISTEEL(I)                                               SFMC  58
000204          IHCON(I) = ISTEEL(I)                                               SFMC  59
000206          CH(I) = CS(I)                                                      SFMC  60
000207          CL(I) = CS(I)                                                      SFMC  61
000211       25 CONTINUE                                                           SFMC  62
          C          CORRECT FOR THE COMPOSITE ACTION                              SFMC  63
000214          IF (JPT3 .GE. 10) GO TO 30                                         SFMC  64
000216          DO 31 I = 1, 10, 2                                                 SFMC  65
000220          IA = LPCM(I)                                                       SFMC  66
000222          IB = LPCM(I+1) - 1                                                 SFMC  67
000224          IF (IA .EQ. 0) GO TO 30                                            SFMC  68
000225          DO 31 J = IA, IB                                                   SFMC  69
000226           ILCON(J) = ISTEEL(J)                                              SFMC  70
000230          IHCON(J) = ISTEEL(J)                                               SFMC  71
000231          CH(J) = CS(J)                                                      SFMC  72
000233          CL(J) = CS(J)                                                      SFMC  73
000234       31 CONTINUE                                                           SFMC  74
000240       30 CONTINUE                                                           SFMC  75
000240          RETURN                                                             SFMC  76
000241          END                                                                SFMC  77

SUBPROGRAM LENGTH
000320

UNUSED COMPILER SPACE
042000



                SUBROUTINE SOLVE (N,L,A,B,NA)                                      SOLV   1
          C          SOLVES THE MATRIX EQUATION FOR THE REACTION INFLUENCE LINES   SOLV   2
000010          DIMENSION A(5,5), B(5,81), MAVE(5,2)                               SOLV   3
000010      201 FORMAT (// 17H SINGULAR MATRIX //)                                 SOLV   4
000010      202 FORMAT (2I13))                                                     SOLV   5
```

```
        C          GAUSS ELIMINATION TO UNIT MATRIX WITH PIVOTING. A IS THE
        C          COEFFICIENT MATRIX SIZE NS BY NS. P IS THE RIGHT HAND SIDE
        C          MATRIX WITH NA COLUMNS AND NS ROWS. EVERYTHING IS SCRAMBLED
        C          IN THE A MATRIX AND THE RESULTS ARE ON TOP OF P. IF A IS
        C          SINGULAR THE PROGRAM STOPS.
        C          ZERO OUT THE REQUIRED VARIABLES
000010        DO 203 I = 1, 5
000011        DO 203 J = 1, 2
000012    203 MAVE (I,J) = 0
000021        IR = 0
000022        IS = 0
000023        NMAX = 0
000024        MMAX = 0
000025        BITTLE = 0.0
000026        INDEX = 0
        C          BEGIN THE SOLUTION LOOP
000027        DO 266 I = 1, N
000030        TS = A(I,I)
000033        NMAX = I
000034        MMAX = I
        C          DETERMINE THE LARGEST ELEMENT
000035        DO 210 J = I, N
000036        DO 210 M = I, N
000040        IF (ABS(A(J,M)) .LT. TS) GO TO 210
000045        TS = ABS(A(J,M))
000050        NMAX = J
000051        MMAX = M
000052    210 CONTINUE
000057        IF (NMAX .EQ. I) GO TO 220
        C          DO THE SIMPLE ROW AND COLUMN OPERATIONS
000061        DO 215 K = I, N
000063        TS = A(NMAX,K)
000066        A(NMAX,K) = A(I,K)
000073    215 A(I,K) = TS
000100        DO 219 K = 1, L
000101        TS = P(NMAX,K)
000104        P(NMAX,K) = P(I,K)
000111    219 P(I,K) = TS
000116    220 IF (MMAX .EQ. I) GO TO 230
000120        DO 225 K = 1, N
000122        TS = A(K,I)
000126        A(K,I) = A(K,MMAX)
000132    225 A(K,MMAX) = TS
000137        INDEX = INDEX + 1
000140        MAVE(INDEX,1) = MMAX
000142        MAVE(INDEX,2) = I
        C          DETERMINE THE SINGULARITY CONDITION
000144    230 IF (ABS(A(I,I)) .GT. BITTLE) GO TO 240
        C          EXIT IF A SINGULAR MATRIX IS FOUND
000153        WRITE (6,201)
000156        WRITE (6,202) I, INDEX
000166        CALL EXIT
        C          CONTINUE THE ROW AND COLUMN OPERATIONS
000167    240 J = N + 1
000174        IF (I .EQ. N) GO TO 243
000175    241 IF (J .EQ. I+1) GO TO 243
000200        J = J - 1
000201        TS = A(I,J)/A(I,I)
```

```
000206              A(I,J) = TS                                                   SOLV   64
000211              GO TO 241                                                     SOLV   65
000211          243 DO 244 J = 1, L                                               SOLV   66
000213              TS = B(I,J)/A(I,I)                                            SOLV   67
000221          244 B(I,J) = TS                                                   SOLV   68
000226          246 IO = I - 1                                                    SOLV   69
000230              IF (IO .EQ. 0) GO TO 257                                      SOLV   70
000231              K = 0                                                         SOLV   71
000232          245 K = K + 1                                                     SOLV   72
000234              TT = A(K,I)                                                   SOLV   73
000240              J = N + 1                                                     SOLV   74
000241              IF (I .EQ. N) GO TO 252                                       SOLV   75
000242          250 IF (J .EQ. I+1) GO TO 252                                     SOLV   76
000245              J = J - 1                                                     SOLV   77
000246              TO = A(K,I)                                                   SOLV   78
000251              TU = A(I,J)                                                   SOLV   79
000254              TV = A(K,J)                                                   SOLV   80
000257              TS = TV - TT * TU                                             SOLV   81
000262              A(K,J) = TS                                                   SOLV   82
000265              GO TO 250                                                     SOLV   83
000266          252 DO 255 J = 1, L                                               SOLV   84
000270              TS = B(K,J) - A(K,I) * B(I,J)                                 SOLV   85
000301          255 B(K,J) = TS                                                   SOLV   86
000306              IF (K .EQ. IO) GO TO 257                                      SOLV   87
000310              GO TO 245                                                     SOLV   88
000310          257 IS = I + 1                                                    SOLV   89
000312              IF (IS .EQ. N+1) GO TO 266                                    SOLV   90
000314              DO 265 K = IS, N                                              SOLV   91
000315              TT = A(K,I)                                                   SOLV   92
000321              IS = I + 1                                                    SOLV   93
000323              IF (IS .EQ. N+1) GO TO 262                                    SOLV   94
000325              J = N + 1                                                     SOLV   95
000326              IF (I .EQ. N) GO TO 262                                       SOLV   96
000327          260 IF (J .EQ. I+1) GO TO 262                                     SOLV   97
000332              J = J - 1                                                     SOLV   98
000333              TO = A(K,I)                                                   SOLV   99
000336              TU = A(I,J)                                                   SOLV  100
000341              TV = A(K,J)                                                   SOLV  101
000344              TS = TV - TT * TU                                             SOLV  102
000347              A(K,J) = TS                                                   SOLV  103
000352              GO TO 260                                                     SOLV  104
000353          262 DO 265 J = 1, L                                               SOLV  105
000355              TS = B(K,J) - A(K,I) * B(I,J)                                 SOLV  106
000366          265 B(K,J) = TS                                                   SOLV  107
000375          266 CONTINUE                                                      SOLV  108
000400              IF (INDEX .EQ. 0) GO TO 280                                   SOLV  109
              C        ADJUST COLUMNS BACK TO THEIR ORIGINAL ORDER                SOLV  110
000401              I = INDEX + 1                                                 SOLV  111
000402          270 IF (I .EQ. 1) GO TO 280                                       SOLV  112
000404              I = I - 1                                                     SOLV  113
000405              J = NAVI(I,2)                                                 SOLV  114
000407              M = NAVI(I,1)                                                 SOLV  115
000410              DO 275 K = 1, L                                               SOLV  116
000412              TS = B(J,K)                                                   SOLV  117
000415              B(J,K) = B(M,K)                                               SOLV  118
000422          275 B(M,K) = TS                                                   SOLV  119
000427              GO TO 270                                                     SOLV  120
000427          280 RETURN                                                        SOLV  121
```

```
000430             END                                                                              SOLV 122

SUBPROGRAM LENGTH
000507

UNUSED COMPILER SPACE
041300


                   SUBROUTINE SORT (PAR,N)                                                          SORT    1
000005             DIMENSION PAR(40), IO(40), P(40)                                                 SORT    2
000005             DO 1 I=1, N                                                                      SORT    3
000006          1  IO(I) = I                                                                        SORT    4
000011             NSTOP = N - 1                                                                    SORT    5
000013             DO 2 K = 1, NSTOP                                                                SORT    6
000014             J = N - K                                                                        SORT    7
000015             ISW = 1                                                                          SORT    8
000016             DO 3 I = 1, J                                                                    SORT    9
000020             IOI = IO(I)                                                                      SORT   10
000021             IOII = IO(I+1)                                                                   SORT   11
000023             IF ( PAR(IOI) .LE. PAR(IOII) ) GO TO 3                                           SORT   12
000030             II = IOI                                                                         SORT   13
000031             IO(I) = IOII                                                                     SORT   14
000032             IO(I+1) = II                                                                     SORT   15
000034             ISW = 2                                                                          SORT   16
000035          3  CONTINUE                                                                         SORT   17
000040             IF (ISW .EQ. 1) GO TO 10                                                         SORT   18
000042          2  CONTINUE                                                                         SORT   19
000044         10  DO 11 I = 1, N                                                                   SORT   20
000046             IA = IO(I)                                                                       SORT   21
000047         11  P(I) = PAR(IA)                                                                   SORT   22
000054             DO 12 I = 1, N                                                                   SORT   23
000055         12  PAR(I) = P(I)                                                                    SORT   24
000061             RETURN                                                                           SORT   25
000061             END                                                                              SORT   26

SUBPROGRAM LENGTH
000226

UNUSED COMPILER SPACE
042700


                   SUBROUTINE STIFF(IDSP,NS,DESREA,NOS,FLTH,FLWID,WEBTH,DEPTH,PROP,                 STIF    1
                  1FM,BEAR)                                                                         STIF    2
              C       DESIGNS THE BEARING STIFFENERS WHERE NEEDED                                   STIF    3
000016             DIMENSION IDSP(5),DESREA(5,4),BL(4),BEAR(5,3),NOS(80),FLTH(18),                  STIF    4
                  1FLWID(18),WEBTH(18),DEPTH(18),PROP(3)                                            STIF    5
000016             DO 420 I = 1, NS                                                                 STIF    6
000017             DO 420 J = 1, 3                                                                  STIF    7
```

```
000020      920 BEAR(I,J) = 0.0                                               STIF   8
          C           BEGIN THE LOOP FOR EACH SUPPORT                          STIF   9
000030          DO 900 I = 1, NS                                              STIF  10
          C           CHECK TO SEE IF A STIFFENER IS REQUIRED                  STIF  11
000031          DO 901 K = 1, 4                                               STIF  12
000032      901 RL(K) = DESREA(I,K)                                           STIF  13
000042          TMR = STRMAX(RL)                                              STIF  14
000045          IB = IOSP(I)                                                  STIF  15
000053          IA = NOS(IB)                                                  STIF  16
000055          IF (I .EQ. NS) IA = NOS(IB - 1)                               STIF  17
000060          SS = TMR/(WEBTH(IA) * (DEPTH(IA) - 2.0 * FLTH(IA)))           STIF  18
000071          IF (ABS(SS) .LT. .75*PROP(2)) GO TO 902                       STIF  19
          C           DESIGN OF THE BEARING STIFFENER                          STIF  20
          C           DETERMINE THE WIDTH OF THE PLATE                         STIF  21
000076          WIDMAX = (FLWID(IA) - WEBTH(IA))/2.0                          STIF  22
000103          WID = 4.0                                                     STIF  23
000104      905 IF ((WID+.25) .GT. WIDMAX) GO TO 904                          STIF  24
000111          WID = WID + .25                                               STIF  25
000112          GO TO 905                                                     STIF  26
          C           DETERMINE THE MINIMUM THICKNESS                          STIF  27
000112      904 TMIN = SQRT(PROP(3)/33.) * WID / 12.                          STIF  28
000121          THICK = .25                                                   STIF  29
000122      907 IF (THICK .GE. TMIN) GO TO 906                                STIF  30
000130          THICK = THICK + .0625                                         STIF  31
000132          GO TO 907                                                     STIF  32
          C           DETERMINE AND CHECK THE BEARING STRESS                   STIF  33
000132      906 BS = TMR/((WID-1.25+WEBTH(IA)/2.)*THICK*2.0)                  STIF  34
000143          IF (BS .LE. .80*PROP(3)) GO TO 908                            STIF  35
000147          THICK = THICK + .0625                                         STIF  36
000151          GO TO 906                                                     STIF  37
          C           DETERMINE AND CHECK THE COLUMN STRESS                    STIF  38
000151      908 COLA = 18.0 * WEBTH(IA)**2 + WID * THICK * 2.0                STIF  39
000160          COLI = 1.5*WEBTH(IA)**4 + (THICK*WID**3)/6.   +               STIF  40
               12.0*WID*THICK*(WID/2.+WEBTH(IA)/2.)**2                        STIF  41
000175          COLR = SQRT(COLI/COLA)                                        STIF  42
000201          SL = (DEPTH(IA) - 2. * FLTH(IA)) * .75/COLR                   STIF  43
000213          FA = PROP(1)*(1.-SL*SL*PROP(3)/(39.47835*EM))/1.25            STIF  44
000223          CSTR = TMR/COLA                                               STIF  45
000225          IF (CSTR .LE. FA) GO TO 909                                   STIF  46
000227          THICK = THICK + .0625                                         STIF  47
000231          GO TO 906                                                     STIF  48
          C           SET UP THE BEAR ARRAY                                    STIF  49
000232      909 BEAR(I,1) = WID                                               STIF  50
000235          BEAR(I,2) = THICK                                             STIF  51
          C           DESIGN THE CONNECTING WELDS                              STIF  52
000237          WL = 4. * (DEPTH(IA) - 2. * FLTH(IA))                         STIF  53
          C           DETERMINE THE ALLOWABLE WELD METAL STRESS                STIF  54
000246          FV = 12.4                                                     STIF  55
000247          IF (ABS(PROP(3) - 36.0) .GT. 1.45) FV = 14.5                  STIF  56
000256          SSK = 1.0 + .36 * (FU/58. - 1.0)                              STIF  57
000263          RP = DESREA(I,1) + DESREA(I,2) + DESREA(I,3)                  STIF  58
000270          RN = DESREA(I,1) + DESREA(I,3) + DESREA(I,4)                  STIF  59
000275          RATIO = RN / RP                                               STIF  60
000277          IF (ABS(RN) .GT. RP) RATIO = RP / RN                          STIF  61
000302          FATS = SSK * 10.8 / (1.0 - .55 * RATIO)                       STIF  62
000307          IF (FATS .LT. FV) FV = FATS                                   STIF  63
          C           CALCULATE THE WELD THICKNESS                             STIF  64
000312          TR = TMR/(.707 * FV * WL)                                     STIF  65
```

```
000316          N16 = TH/.0625                                                  STIF  66
000321          BEAR(I,3) = N16 * .0625 + .0625                                 STIF  67
000325          IF (BEAR(I,3) .LT. .25) BEAR(I,3) = .25                         STIF  68
000331          GO TO 900                                                       STIF  69
         C         SET UP THE BEAR ARRAY FOR NO STIFFENER                       STIF  70
000332      902 BEAR(I,1) = 0.0                                                 STIF  71
000335          BEAR(I,2) = 0.0                                                 STIF  72
000336          BEAR(I,3) = 0.0                                                 STIF  73
000337      900 CONTINUE                                                        STIF  74
000342          RETURN                                                          STIF  75
000342          END                                                             STIF  76

SUBPROGRAM LENGTH
000477

UNUSED COMPILER SPACE
041500




                SUBROUTINE STRCHK                                               STRC   1
         C         CALCULATES THE STRESSES AT EACH ANALYSIS POINT              STRC   2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), STRC   3
               1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),      STRC   4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,   STRC   5
               3SLABWD,SLABTH,MR,COOR(81),RIL(5,81,3),CC(5,5),FM,SPAN(4),IOSP(5), STRC   6
               4TMOFI(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLO(10),PROP(3),STRC   7
               5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),BEAR(5,3),STRC   8
               6CONST(9),CONSP(16,3,4),COVPL(12,6,2)                           STRC   9
000002          COMMON/ONE/JPT3,IPT10                                           STRC  10
000002          COMMON/TWO/LPCM                                                 STRC  11
000002          COMMON/SIX/FU                                                   STRC  12
000002          COMMON/TEN/HAUNCH                                               STRC  13
000002          REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                         STRC  14
000002          DIMENSION LPCM(10), TMST(4)                                     STRC  15
000002          IF (JPT3 .LT. 9) TSWID1 = SLABWD / (3. * MR)                    STRC  16
000010          IF (JPT3 .LT. 9) TSWID2 = SLABWD / MR                           STRC  17
000014          DO 1800 I = 1, NA                                               STRC  18
000016          DO 1800 J = 1, 4                                                STRC  19
000017          DO 1800 K = 1, 2                                                STRC  20
000020     1800 STRESS(I,J,K) = 0.0                                             STRC  21
         C         BEGIN THE LOOP TO CALCULATE THE STRESSES                     STRC  22
000034          DO 800 I = 1, NA                                                STRC  23
000036          PLTI = 0.0                                                      STRC  24
000037          PLWI = 0.0                                                      STRC  25
000040          PLTJ = 0.0                                                      STRC  26
000041          PLWJ = 0.0                                                      STRC  27
         C         DETERMINE THE PLATE THICKNESS AT EACH NODE                   STRC  28
000042          DO 803 K = 1, 2                                                 STRC  29
000043          DO 801 J = 1, 12                                                STRC  30
000044          IF (ABS(COVPL(J,1,K)).LT..0001.AND.ABS(COVPL(J,2,K)).LT..0001)  STRC  31
               1GO TO 803                                                       STRC  32
000064          IF (COOR(I).GE.COVPL(J,1,K).AND.COOR(I).LE.COVPL(J,2,K)) GO TO 802STRC  33
000077          GO TO 801                                                       STRC  34
000077      802 IF (K .EQ. 2) GO TO 810                                         STRC  35
```

```
000101            PLTT = COVPL(J,3,K)
000105            PLWT = COVPL(J,4,K)
000110            GO TO 803
000110        810 PLTB = COVPL(J,3,K)
000114            PLWB = COVPL(J,4,K)
000117            GO TO 803
000120        801 CONTINUE
000122        803 CONTINUE
         C             CALCULATE THE SECTION PROPERTIES
000124            IA = I
000126            IF (I .EQ. NA) GO TO 821
000130            IF (ABS(PLTT-PLATH(IA,1)).LT..0001.AND.ABS(PLTB-PLATH(IA,2))
                 1.LT..0001) GO TO 820
000146            IF (ABS(PLTT-PLATH(IA-1,1)).LT..0001.AND.ABS(PLTB-PLATH(IA-1,2))
                 1.LT..0001) GO TO 821
000164            JJ = NOS(IA)
000166            HP = DEPTH(JJ)/2. + PLTB
000171            CALL ICALC (PLTT,PLWT,PLTB,PLWB,DEPTH(JJ),SAREA(JJ),IX(JJ),HP,0.,
                 1SC,SIS)
000205            IF (JPT3 .GT. 9) GO TO 830
000211            IF (LESMOM(I,1) .LT. 0.0) GO TO 830
000213            H = 2. * (DEPTH(JJ) + PLTB - SC)
000217            TSA = SAREA(JJ) + PLTT*PLWT + PLTB*PLWB
000225            CALL ICALC(TSWID1,SLABTH,0.,0.,H,TSA,SIS,SC,HAUNCH,SL,SLS)
000237            CALL ICALC (TSWID2,SLABTH,0.,0.,H,TSA,SIS,SC,HAUNCH,SH,SHS)
000252            GO TO 850
000253        821 IA = I - 1
000255        820 SIS = TSTEEL(IA)
000257            SC = CS(IA)
000261            IF (JPT3 .GT. 9) GO TO 830
000264            IF (LESMOM(I,1) .LT. 0.0) GO TO 830
000266            SLS = ILCON(IA)
000270            SL = CL(IA)
000271            SHS = IHCON(IA)
000273            SH = CH(IA)
000274            GO TO 850
000275        830 SLS = SIS
000277            SL = SC
000300            SHS = SIS
000301            SH = SC
000302        850 IS = NOS(IA)
000304            TOTD = DEPTH(IS) + PLTT + PLTB
         C             CALCULATE THE BOTTOM FIBER STRESS
000307            TMST(1) = DESMOM(I,1) * SC / SIS
000312            TMST(2) = DESMOM(I,2) * SH / SHS
000315            TMST(3) = DESMOM(I,3) * SL / SLS
000317            TMST(4) = DESMOM(I,4) * SH / SHS
000322            TBS = STRMAX(TMST)
000324            STRESS(I,1,1) = TBS
         C             DETERMINE THE ALLOWABLE BOTTOM FIBER STRESSES
000326            ASTR = ALLOW(DESMOM(I,1),FLWID(IS),LPCM,TYSP,I,NS,PROP,COOR,
                 1DESMOM,SPNLEN,EM,1)
000343            IF (I .EQ. 1) GO TO 901
000345            IF (I .EQ. NA) GO TO 901
000347            AFS = BOSFAT(TMST,FT,STRESS(I,1,1),PROP(2))
000353            IF (AFS .LT. ASTR) ASTR = AFS
000356        901 STRESS(I,1,2) = 0.0
000360            IF (ABS(STRESS(I,1,1)).GT.ABS(ASTR)) STRESS(I,1,2) = 1.0
```

```
              C          CALCULATE THE TOP STRESS                                        STRC  94
000367                  TMST(1) = DESMOM(I,1) * (TOTD - SC) / SIS                        STRC  95
000373                  TMST(2) = DESMOM(I,2) * (TOTD - SH) / SHS                        STRC  96
000377                  TMST(3) = DESMOM(I,3) * (TOTD - SL) / SLS                        STRC  97
000403                  TMST(4) = DESMOM(I,4) * (TOTD - SH) / SHS                        STRC  98
000407                  TDS = STRMAX(TMST)                                               STRC  99
000412                  STRESS(I,2,1) = -1.0 * TDS                                       STRC 100
              C          DETERMINE THE ALLOWABLE TOP FIBER STRESSES                      STRC 101
000414                  ASTR = PROP(1)                                                   STRC 102
000416                  IF (I .EQ. 1) GO TO 903                                          STRC 103
000420                  IF (I .EQ. NA) GO TO 903                                         STRC 104
000422                  ASE = HASEAT(TMST,FJ,STRESS(I,2,1),PROP(3))                      STRC 105
000426                  IF (ASE .LT. ASTR) ASTR = AES                                    STRC 106
000431              903 STRESS(I,2,2) = 0.0                                              STRC 107
000433                  IF (ABS(STRESS(I,2,1)).GT.ASTR) STRESS(I,2,2) = 1.0              STRC 108
              C          CALCULATE THE CONCRETE STRESSES IF NECESSARY                    STRC 109
000440                  IF (JPT3 .GT. 9) GO TO 853                                       STRC 110
000444                  IF (DESMOM(I,1) .LT. 0.0) GO TO 853                              STRC 111
000445                  TOTD = TOTD - FLTT + HAUNCH + SLABTH                             STRC 112
000451                  TMST(1) = 0.0                                                    STRC 113
000452                  TMST(2) = DESMOM(I,2) * (TOTD - SH) / (SHS * MR)                 STRC 114
000460                  TMST(3) = DESMOM(I,3) * (TOTD - SL) / (SLS * 3. * MR)            STRC 115
000466                  TMST(4) = DESMOM(I,4) * (TOTD - SH) / (SHS * MR)                 STRC 116
000474                  TDS = STRMAX(TMST)                                               STRC 117
000476                  STRESS(I,3,1) = -1.0 * TDS                                       STRC 118
000500                  STRESS(I,3,2) = 0.0                                              STRC 119
000501                  IF (ABS(STRESS(I,3,1)) .GT. FC) STRESS(I,3,2) = 1.0              STRC 120
000506                  GO TO 870                                                        STRC 121
000507              853 STRESS(I,3,1) = 0.0                                              STRC 122
000511                  STRESS(I,3,2) = 0.0                                              STRC 123
              C          CALCULATE THE SHEAR STRESSES                                    STRC 124
000512              870 JJ = NOS(IA)                                                     STRC 125
000514                  AWEB = DEPTH(JJ) * WEBTH(JJ)                                     STRC 126
000516                  IF (I .EQ. 1) GO TO 860                                          STRC 127
000520                  IF (I .EQ. NA) GO TO 861                                         STRC 128
000522                  IR = 2 * I - 1                                                   STRC 129
000523                  IL = IR - 1                                                      STRC 130
000524                  GO TO 880                                                        STRC 131
000525              860 IR = 1                                                           STRC 132
000526                  T2 = 0.0                                                         STRC 133
000527                  GO TO 880                                                        STRC 134
000530              861 IL = 2 * NA - 2                                                  STRC 135
000532                  T1 = 0.0                                                         STRC 136
000533                  GO TO 888                                                        STRC 137
000534              880 DO 865 K = 1, 4                                                  STRC 138
000536              865 TMST(K) = DESSH(IR,K) / AWEB                                     STRC 139
000545                  T1 = STRMAX(TMST)                                                STRC 140
000550              888 DO 866 K = 1, 4                                                  STRC 141
000552              866 TMST(K) = DESSH(IL,K) / AWEB                                     STRC 142
000561                  T2 = STRMAX(TMST)                                                STRC 143
000564                  STRESS(I,4,1) = ABS(T1)                                          STRC 144
000567                  IF (ABS(T2) .GT. ABS(T1)) STRESS(I,4,1) = ABS(T2)                STRC 145
000575                  STRESS(I,4,2) = 0.0                                              STRC 146
000577                  IF (ABS(STRESS(I,4,1)) .GT. PROP(2)) STRESS(I,4,2) = 1.0         STRC 147
000604              890 CONTINUE                                                         STRC 148
000607                  RETURN                                                           STRC 149
000607                  END                                                              STRC 150
```

```
SUBPROGRAM LENGTH
000735

UNUSED COMPILER SPACE
040100


                   FUNCTION STRMAX (ISSS)                                     STRM    1
            C         DETERMINES THE DESIGN STRESS AT A POINT GIVEN           STRM    2
            C         THE COMPONENT DESIGN STRESSES FOR THAT POINT            STRM    3
000003             DIMENSION ISSS(4)                                          STRM    4
000003             TEMP = ISSS(1) + ISSS(3)                                   STRM    5
000004             PAR1 = TEMP + ISSS(2)                                      STRM    6
000006             PAR2 = TEMP + ISSS(4)                                      STRM    7
000010             ANS = PAR1                                                 STRM    8
000012             IF (ABS(PAR2) .GT. ABS(ANS)) ANS = PAR2                    STRM    9
000017             STRMAX = ANS                                               STRM   10
000020             RETURN                                                     STRM   11
000021             END                                                        STRM   12

SUBPROGRAM LENGTH
000043

UNUSED COMPILER SPACE
043100


                   SUBROUTINE TERMIN                                          TERM    1
            C         DETERMINE THE CUTOFF DISTANCE FOR A GIVEN PLATE         TERM    2
000002             COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), TERM    3
                  1DAF(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),  TERM    4
                  2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,  TERM    5
                  3SLABWD,SLABTH,MP,COOR(81),BIL(5,81,3),CC(5,5),FI,SPAN(4),IOSP(5), TERM    6
                  4TMOM1(80),NA,LENGTH,RU(81),SUBLEN(93),NS,UNLO(9),TRLO(10),PROP(3),TERM    7
                  5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),REAR(5,3),TERM    8
                  6COST(9),CONSP(16,3,4),COVPL(12,6,2)                        TERM    9
000002             COMMON/ONE/JPT3,IPT10                                      TERM   10
000002             COMMON/TWO/LPCM                                            TERM   11
000002             COMMON/SIX/FU                                              TERM   12
000002             COMMON/EIGHT/IST                                           TERM   13
000002             COMMON/TEN/HAUNCH                                          TERM   14
000002             COMMON/ELEVEN/KD,KC,SIS,SHS,SLS,ISS, IJ,SS,DMAX,DIST       TERM   15
000002             COMMON/TEEN/PTH,KT,ITT                                     TERM   16
000002             REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH,TLT                TERM   17
000002             DIMENSION LPCM(10), TMST(4), PTH(40,4)                     TERM   18
            C         INITIALIZE THE REQUIRED VARIABLES                       TERM   19
000002             MN = NS - 1                                                TERM   20
000004             TTN = 0.0                                                  TERM   21
000005             KLL = 0                                                    TERM   22
000006             KTT = 0                                                    TERM   23
000007             DP = 0.0                                                   TERM   24
```

```
000010           AV = 9999999.                                                  TERM  25
          C            DETERMINE THE MOMENT AND STRESS AT THE POINT             TERM  26
000011           DO 5801 I = 2, NN                                              TERM  27
000013           IF (IOSP(I) .EQ. 0) GO TO 5802                                 TERM  28
000014           IB = IOSP(I)                                                   TERM  29
000016           IF (ABS(COOR(KQ)-COOR(IB)).GT.AV) GO TO 5801                   TERM  30
000024           IA = IOSP(I)                                                   TERM  31
000026           IF (KQ .LE. IA) IA = IA - 1                                    TERM  32
000032           AV = ABS(COOR(KQ) - COOR(IB))                                  TERM  33
000036      5801 CONTINUE                                                       TERM  34
000041      5802 CONTINUE                                                       TERM  35
000041           IF (NN .EQ. 1) IA = KQ                                         TERM  36
000045           T1 = 0.0                                                       TERM  37
000046           DO 5600 K = 1, 4                                               TERM  38
000050      5600 TMST(K) = DESMOM(KQ,K)                                         TERM  39
000057           GO TO 5601                                                     TERM  40
000057      5500 DO 5051 K = 1, 4                                               TERM  41
000061           DO 5052 KK = 1, NA                                             TERM  42
000062      5052 RO(KK) = DESMOM(KK,K)                                          TERM  43
000071      5051 TMST(K) = ILINT(KQ,RO,COOR,ISW,DP,LENGTH,SUBLEN)               TERM  44
000104      5601 CONTINUE                                                       TERM  45
000104           DMS = TMST(1)                                                  TERM  46
000106           IF (DMS .LT. 0.0) GO TO 5700                                   TERM  47
000107           TMST(1) = TMST(1) / SIS                                        TERM  48
000110           TMST(2) = TMST(2) / SHS                                        TERM  49
000112           TMST(3) = TMST(3) / SLS                                        TERM  50
000113           TMST(4) = TMST(4) / SHS                                        TERM  51
000114           GO TO 5701                                                     TERM  52
000115      5700 TMST(1) = TMST(1) / SS                                         TERM  53
000117           TMST(2) = TMST(2) / SS                                         TERM  54
000120           TMST(3) = TMST(3) / SS                                         TERM  55
000121           TMST(4) = TMST(4) / SS                                         TERM  56
000122      5701 TDS = STRMAX(TMST)                                             TERM  57
          C           DETERMINE THE ALLOWABLE STRESSES                          TERM  58
000125           ASTRT = PROP(1)                                                TERM  59
000126           ASTRB = ALLOW(DMS,FLWID(IST),LPCM,IOSP,IA,NS,PROP,COOR,DESMOM, TERM  60
                1SUBLEN,FM,2)                                                   TERM  61
000143           SMIN = TMST(1) + TMST(2) + TMST(3)                             TERM  62
000146           SMAX = TMST(1) + TMST(3) + TMST(4)                             TERM  63
000150           IF (ABS(SMIN) .GT. .0001 .AND. ABS(SMAX) .GT. .0001) GO TO 5100 TERM  64
000164           R = 0.0                                                        TERM  65
000165           GO TO 5110                                                     TERM  66
000165      5100 RATIO = SMIN / SMAX                                            TERM  67
000167           IF (ABS(SMAX) .LT. ABS(SMIN)) RATIO = SMAX / SMIN              TERM  68
000173      5110 SK = 1.0 + .23 * (FU/58. - 1.0)                                TERM  69
000200           IF (KC .EQ. 2) GO TO 5060                                      TERM  70
000202           ASFT = 12. / (1.0 - RATIO)                                     TERM  71
000204           ASFB = ASFT                                                    TERM  72
000205           GO TO 5061                                                     TERM  73
000206      5060 ASFT = .55*PROP(3)/(1.-(.55*PROP(3)/(SK*13.6)-1.)*RATIO)       TERM  74
000216           ASFB = SK * 17.2 / (1. - .62 * RATIO)                          TERM  75
000223           IF (DMS .GT. 0.0) GO TO 5061                                   TERM  76
000225           TEMP = ASFT                                                    TERM  77
000226           ASFT = ASFB                                                    TERM  78
000227           ASFB = TEMP                                                    TERM  79
000230      5061 IF (JJ .EQ. 2) GO TO 6080                                      TERM  80
000232           AS = ASTRT                                                     TERM  81
000234           IF (ASFT .LT. ASTRT) AS = ASFT                                 TERM  82
```

```
000237              GO TO 6001                                                          TERM   83
000240        6000  AS = ASIN                                                           TERM   84
000242              IF (ASE  .LT. ASTAR) AS = ASEM                                      TERM   85
            C              START CUTTING OFF THE PLATE BY USING AN INCREASING           TERM   86
            C              STEP SIZE OPTIMIZATION PROCEDURE                             TERM   87
000245        6001  IF (ABS(DP) .LT. .00001) GO TO 5056                                 TERM   88
000252              IF (ABS(TOS) .LT. AS) GO TO 5057                                    TERM   89
000255              GO TO 5058                                                          TERM   90
000255        5057  ISW = ISS                                                           TERM   91
000257              GO TO 5059                                                          TERM   92
000257        5058  ISW = 1                                                             TERM   93
000260              KLL = 1                                                             TERM   94
000261              IF (ISS .EQ. 1) ISW = 2                                             TERM   95
000264              GO TO 5059                                                          TERM   96
000265        5056  IF (ABS(TOS).LT.AS.AND.KLL.EQ.0) GO TO 6500                         TERM   97
000276              IF (ABS(TOS).GT.AS.AND.KLL.EQ.1) GO TO 6030                         TERM   98
000307              GO TO 5082                                                          TERM   99
000307        5059  BL = 0.                                                             TERM  100
000310              YK = 1.                                                             TERM  101
000312              IF (ISW .EQ. 2) YK = -1.0                                           TERM  102
000315              VDS = 99999.                                                        TERM  103
000317              V = .5                                                              TERM  104
000320        5090  IF (ABS(DP-DMAX) .LT. .0001) GO TO 6000                             TERM  105
000325        5100  V = V * 2.                                                          TERM  106
000327              TT = DP                                                             TERM  107
000330              DP = BL + V                                                         TERM  108
000332              IF (DP .GT. DMAX) DP = DMAX                                         TERM  109
000335              IF (ABS(DP - VDS) .LT. .00001) GO TO 5082                           TERM  110
000341              GO TO 5595                                                          TERM  111
000342        5082  IF (DP .LT. VDS) VDS = DP                                           TERM  112
000345              IF (ABS(BL - VDS) .LT. 1.5) GO TO 5095                              TERM  113
000352              BL = TT                                                             TERM  114
000353              V = .5                                                              TERM  115
000354              GO TO 5100                                                          TERM  116
000355        6030  BL = DP                                                             TERM  117
000357        5095  DIST = YK * BL                                                      TERM  118
000361              IF(ABS(ABS(DIST)-DMAX).GT.1) GO TO 6500                             TERM  119
000370              IF (KTT .EQ. 1) GO TO 6500                                          TERM  120
000372              IF (ABS(TOS) .LT. AS) GO TO 6500                                    TERM  121
000375              BL = DP                                                             TERM  122
            C              ADJUST DMAX IF REQUIRED                                      TERM  123
000377              TTN = TTN + 1.0                                                     TERM  124
000401              DMAX = DMAX + DMAX/TTN                                              TERM  125
000403              IF (KI .EQ. 1) GO TO 7000                                           TERM  126
000405              IF (DMAX .GT. COOR(KO)-PTH(ITT-1,3)) GO TO 7001                     TERM  127
000413              GO TO 7002                                                          TERM  128
000413        7001  DMAX = COOR(KO) - PTH(ITT-1,3)                                      TERM  129
000416              KTT = 1                                                             TERM  130
000417              GO TO 7002                                                          TERM  131
000420        7000  IF (DMAX .GT. PTH(ITT+1,4)-COOR(KO)) GO TO 7003                     TERM  132
000426              GO TO 7002                                                          TERM  133
001426        7003  DMAX = PTH(ITT+1,4) - COOR(KO)                                      TERM  134
000431              KTT = 1                                                             TERM  135
000432        7002  IF (IPTIO .EQ. 6) WRITE (6,1) DMAX                                  TERM  136
000442           1  FORMAT (13X,7HDMAX = ,F15.5)                                        TERM  137
000442              V = .5                                                              TERM  138
000444              GO TO 5100                                                          TERM  139
000444        6500  CONTINUE                                                            TERM  140
```

```
000444          RETURN                                                          TERM 141
000445          END                                                             TERM 142

SUBPROGRAM LENGTH
000567

UNUSED COMPILER SPACE
040400


                SUBROUTINE THDES                                                THDE   1
          C         DETERMINE THE REQUIRED THICKNESSES IN EACH SUBELEMENT       THDE   2
000002          COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), THDE   3
               1DAF(18),IX(18),IY(18),NSECT(18),NDS(80),ISTEEL(80),CS(80),      THDE   4
               2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,   THDE   5
               3SLABWD,SLABTH,MR,COOR(81),RTL(5,81,3),CC(5,5),F(5),SPAN(4),TOSP(5), THDE   6
               4TOREL(81),NA,LENGTH,RD(81),SUBLEN(80),NS,UNLD(9),TPLO(10),PROP(3),THDE   7
               5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESSH(162,4),REAR(5,3),THDE   8
               6COST(9),CONSP(16,3,4),COVPL(12,6,2)                             THDE   9
000002          COMMON/ONE/JPT3,IPT10                                           THDE  10
000002          COMMON/TWO/LPCW                                                 THDE  11
000002          COMMON/SIX/FU                                                   THDE  12
000002          COMMON/EIGHT/IST                                                THDE  13
000002          COMMON/NINE/ISPLIT                                              THDE  14
000002          COMMON/TEN/HAUNCH                                               THDE  15
000002          REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                         THDE  16
000002          DIMENSION IMST(4), LPCM(10)                                     THDE  17
          C         ZERO OUT THE DESIGN ARRAYS                                  THDE  18
000002          DO 8600 I = 1, NF                                               THDE  19
000004          DO 8600 J = 1, 2                                                THDE  20
000005          PLATH(I,J) = 0.0                                                THDE  21
000010     8600 PLWID(I,J) = 0.0                                                THDE  22
000020          IF (JPT3 .LT. 9) TSWID1 = SLABWD / (3. * MR)                    THDE  23
000025          IF (JPT3 .LT. 9) TSWID2 = SLABWD / MR                           THDE  24
000031          PLW = 10.0                                                      THDE  25
000033          IF (FLWID(IST) .GT. 13.0) PLW = 14.0                            THDE  26
000037          IF (FLWID(IST) .GT. 16.0) PLW = 15.0                            THDE  27
          C         BEGIN THE LOOP FOR EACH ELEMENT                             THDE  28
000044          DO 8900 I = 1, NF                                               THDE  29
          C         DETERMINE CRITICAL MOMENT LOCATION IN THE ELEMENT           THDE  31
000046          ISW13 = 0                                                       THDE  30
000047          IF (DESMOM(I,1).LT.0.0.OR.DESMOM(I+1,1).LT.0.0) GO TO 8500      THDE  32
000056          KEY = 0                                                         THDE  33
000057          T1 = DESMOM(I,1) + DESMOM(I,2) + DESMOM(I,3)                    THDE  34
000062          T2 = DESMOM(I+1,1) + DESMOM(I+1,2) + DESMOM(I+1,3)              THDE  35
000065          GO TO 8501                                                      THDE  36
000065     8500 KEY = 1                                                         THDE  37
000066          IF (DESMOM(I,1).GE.0.0 .OR. DESMOM(I+1,1).GE.0.0) GO TO 8502    THDE  38
000077          T1 = DESMOM(I,1) + DESMOM(I,3) + DESMOM(I,4)                    THDE  39
000102          T2 = DESMOM(I+1,1) + DESMOM(I+1,3) + DESMOM(I+1,4)              THDE  40
000105     8501 IA = I                                                          THDE  41
000107          IF (ABS(T1) .LT. ABS(T2)) IA = I + 1                            THDE  42
000115          GO TO 8503                                                      THDE  43
000116     8502 IA = I                                                          THDE  44
```

```
000120          IF (DESMOM(I+1,1) .LT. 0.0) IA = I + 1
          C        START THE PLATE DETERMINATION
000123     8513 CONTINUE
000123     8511 KP = 0
000124          IF (DESMOM(IA,1) .LT. 0.0) DSMN = DESMOM(IA,1) + DESMOM(IA,3) +
               1DESMOM(IA,4)
          C        START WITH NO PLATES
000131          PLTT = 0.0
000132          PLTB = 0.0
          C        DETERMINE THE SECTION PROPERTIES
000133     8545 HP = DEPTH(IST)/2. + PLTB
000137          CALL ICALC (PLW,PLTT,PLW,PLTB,DEPTH(IST),SAREA(IST),TX(IST),HP,
               10.0,SECC,SECI)
000153          IF (DESMOM(IA,1) .LT. 0.0) GO TO 8540
000155          IF (JPT3 .GT. 9) GO TO 700
000161          TSA = SAREA(IST) + PLW * (PLTT + PLTB)
000165          H = 2. * (DEPTH(IST) + PLTB - SECC)
000171          CALL ICALC(TSWID1,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECL,SECIL)
000204          CALL ICALC(TSWID2,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECH,SECIH)
000217          TD = DEPTH(IST) + PLTB + PLTT
000223          TDI = TD - PLTT + SLABTH + HAUNCH
000226          GO TO 701
000227      700 SECH = SECC
000231          SECL = SECC
000232          SECIH = SECI
000233          SECIL = SECI
000234      701 CONTINUE
          C        DETERMINE THE BOTTOM STRESSES IN POSITIVE SECTION
000234          TMST(1) = DESMOM(IA,1) * SECC / SECT
000237          TMST(2) = DESMOM(IA,2) * SECH / SECIH
000242          TMST(3) = DESMOM(IA,3) * SECL / SECIL
000244          TMST(4) = DESMOM(IA,4) * SECH / SECIH
          C        DETERMINE THE ALLOWABLE STRESS
000247          AFS = BASFAT(TMST,FJ,1000.,PROP(3))
000253          ASTRB = PROP(1)
000254          IF (ABS(AFS) .LT. ASTRB) ASTRB = ABS(AFS)
000261          TDS = STRMAX(TMST)
000264          IF (ABS(TDS) .LE. ABS(ASTRB)) GO TO 8541
000270          GO TO 8542
          C        DETERMINE THE BOTTOM STRESS IN NEGATIVE REGION
000271     8540 EFS = DSMN * SECC / SECT
000274          DO 8661 KI = 1, 4
000275     8661 TMST(KI) = DESMOM(IA,KI)
          C        DETERMINE THE ALLOWABLE STRESS
000304          AFSB = BASFAT(TMST,FU,-1000.,PROP(3))
000310          ASTRB = ALLOW(DESMOM(IA,1),FLWID(IST),LPCM,TOSP,IA,NS,PROP,COOR,
               1DESMOM,SUBLEN,FU,1)
000325          IF (ABS(AFSB) .LT. ASTRB) ASTRB = ABS(AFSB)
000331          AFST = BASFAT(TMST,FU,1000.,PROP(3))
000335          ASTRT = PROP(1)
000336          IF (ABS(AFST) .LT. ASTRT) ASTRT = ABS(AFST)
000343          IF (ABS(EFS) .GT. ABS(ASTRB)) GO TO 8542
000350          IF (ABS(EFS) .GT. ABS(ASTRT)) GO TO 8542
000355          GO TO 8541
          C        USE INTERVAL HALVING TO VARY THE PLATE SIZE
000355     8542 IF (KP .EQ. 1) GO TO 8700
000357          IF (ABS(PLTB) .LT. .0001) GO TO 8560
000363          IF (ABS(PLTB) .GT. 1.5 * FLTH(IST)) GO TO 8590
```

```
000370          BL = PLTB                                                        THDE 103
000371          KS = 1                                                           THDE 104
000372          GO TO 8565                                                       THDE 105
000372    8560  UL = 2.1875                                                      THDE 106
000374          BL = .1875                                                       THDE 107
000375          IF (IST .LT. 3) UL = 4.1875                                      THDE 108
000401          GO TO 8565                                                       THDE 109
000402    8581  IF (KP .EQ. 1) GO TO 8701                                        THDE 110
000404          IF (ABS(PLTB) .LT. .0001) GO TO 8591                             THDE 111
000410          UL = PLTB                                                        THDE 112
000411          KS = 0                                                           THDE 113
000412    8565  TH = (UL + BL) / 2.                                              THDE 114
000415          IF (ABS(UL - TH) .LT. .0625) GO TO 8582                          THDE 115
000421          PLTB = TH                                                        THDE 116
000422          PLTT = TH                                                        THDE 117
000423          IF (JPT3 .GT. 9) GO TO 8545                                      THDE 118
000426          IF (LESMOM(IA,1) .GE. 0.0) PLTT = 0.0                            THDE 119
000431          GO TO 8545                                                       THDE 120
000432    8582  IF (KS .EQ. 0) GO TO 8587                                        THDE 121
000433          PLTB = PLTB + .0625                                              THDE 122
000435          KP = 1                                                           THDE 123
000436    8587  IF (PLTB .GT. 1.5 * FLTH(IST)) GO TO 8580                        THDE 124
        C             STORE THE THICKNESS REQUIRED IN THE PROPER ARRAY           THDE 125
000443          PLATH(I,2) = PLTB                                                THDE 126
000444          PLWID(I,2) = PLW                                                 THDE 127
000446          IF (JPT3 .GT. 9) GO TO 80                                        THDE 128
000451          IF (LESMOM(IA,1) .GE. 0.0) GO TO 8595                            THDE 129
000453      80  PLATH(I,1) = PLTB                                                THDE 130
000455          PLWID(I,1) = PLW                                                 THDE 131
000457          GO TO 8596                                                       THDE 132
000457    8591  DO 8599 J = 1, 2                                                 THDE 133
000461          PLATH(I,J) = 0.0                                                 THDE 134
000464    8599  PLWID(I,J) = 0.0                                                 THDE 135
000471          IF (LESMOM(IA,1) .LT. 0.0) GO TO 8900                            THDE 136
000473          GO TO 8596                                                       THDE 137
000474    8595  PLATH(I,1) = 0.0                                                 THDE 138
000476          PLWID(I,1) = 0.0                                                 THDE 139
000477          GO TO 8596                                                       THDE 140
000477    8580  CONTINUE                                                         THDE 141
000477          ISWIT = 1                                                        THDE 142
000500          IF (IPT1 .EQ. 4) WRITE (6,7000) NAME(IST)                        THDE 143
000510    7000  FORMAT (///10X,4HTHE ,A7,25H SECTION CANNOT BE USED.)            THDE 144
000510          RETURN                                                           THDE 145
000511    8701  PLATH(I,1) = PLTT                                                THDE 146
000513          PLATH(I,2) = PLTB                                                THDE 147
000515          PLWID(I,1) = PLW                                                 THDE 148
000516          PLWID(I,2) = PLW                                                 THDE 149
000517    8596  IF (KS .EQ. 1 .AND. KP .NE. 1) GO TO 8597                        THDE 150
        C             COMPUTE THE SECTION PROPERTIES IF REQUIRED                 THDE 151
000526          HP = DEPTH(IST)/2. + PLATH(I,2)                                  THDE 152
000532          CALL ICALC (PLWID(I,1),PLATH(I,1),PLWID(I,2),PLATH(I,2),         THDE 153
     .         1DEPTH(IST),SAREA(IST),IX(IST),HP,0.,0.,SECC,SECT)                THDE 154
000546          KS = 0                                                           THDE 155
000547          IF (JPT1 .GT. 9) GO TO 702                                       THDE 156
000553          TSA = SAREA(IST) + PLWID(I,1)*PLATH(I,1) + PLWID(I,2)*PLATH(I,2) THDE 157
000561          H = 2. * (DEPTH(IST) + PLATH(I,2) - SECC)                        THDE 158
000566          CALL ICALC(ISWID1,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECL,SECIL) THDE 159
000600          CALL ICALC(ISWID2,SLABTH,0.,0.,H,TSA,SECT,SECC,HAUNCH,SECH,SECIH) THDE 160
```

```
000613          TD = DEPTH(IST) + PLATH(I,1) + PLATH(I,2)
000617          TDT = TD - PLATH(I,1) + SLABTH + HAUNCH
000623          GO TO 8547
000624      702 SECL = SECC
000626          SECH = SECC
000627          SECIL = SECI
000630          SECIH = SECI
000631          GO TO 8547
000631     8710 IF (ABS(PLTB) .LT. .0001) PLTB = .1875
000636          PLTB = PLTB + .0625
000640          IF (PLTB .LE. 1.5 * FLTH(IST)) GO TO 8545
000644          ISWIT = 1
000645          RETURN
000646     8547 IF (JPT3 .GT. 9) GO TO 8900
          C        DETERMINE THE TOP STRESSES
000652          TMST(1) = DESMOM(IA,1) * (TD - SECC) / SECI
000656          TMST(2) = DESMOM(IA,2) * (TD - SECH) / SECIH
000662          TMST(3) = DESMOM(IA,3) * (TD - SECL) / SECIL
000666          TMST(4) = DESMOM(IA,4) * (TD - SECH) / SECIH
          C        DETERMINE THE ALLOWABLE STRESS
000672          AFS = BASFAT(IMST,FI,-1000.,PROP(3))
000676          ASTRT = PROP(1)
000677          IF (ABS(AFS) .LT. ASTRT) ASTRT = ABS(AFS)
000704          TDS = STRMAX(TMST)
000707          IF (ABS(TDS) .LE. ASTRT) GO TO 6060
          C        INCREASE THE TOP PLATE THICKNESS IF REQUIRED
000712          IF (ABS(PLTT) .LT. .0001) PLTT = .1875
000716          KP = 1
000717          PLTT = PLTT + .0625
000721          IF (PLTT .LE. 1.5 * FLTH(IST)) GO TO 8545
000725          ISWIT = 1
000726          IF (IPT10 .EQ. 4) WRITE (6,7000) NAME(IST)
000735          RETURN
000736     6060 IF (JPT3 .GT. 9) GO TO 8900
          C        CALCULATE THE CONCRETE STRESS
000742          TMST(1) = 0.0
000742          TMST(2) = DESMOM(IA,2) * (TDT - SECH) / (SECTH * MR)
000750          TMST(3) = DESMOM(IA,3) * (TDT - SECL) / (SECTL * 3. * MR)
000757          TMST(4) = DESMOM(IA,4) * (TDT - SECH) / (SECTH * MR)
000764          TDS = STRMAX(TMST)
000767          IF (ABS(TDS) .LE. FC) GO TO 8900
          C        INCREASE THE TOP PLATE IF REQUIRED
000772          IF (ABS(PLTT) .LT. .0001) PLTT = .1875
000775          KP = 1
000777          PLTT = PLTT + .0625
001001          IF (PLTT .LE. 1.5 * FLTH(IST)) GO TO 8545
001005          ISWIT = 1
001006          IF (IPT10 .EQ. 4) WRITE (6,7000) NAME(IST)
001015          RETURN
001016     8900 CONTINUE
          C        PRINT THE PLATE SIZES IF REQUIRED BY IPT1
001021          IF (IPT10 .NE. 4) GO TO 7200
001023          WRITE (6,7100)
001026     7100 FORMAT(1H1,25X,4HSECT,4X,3HTOP,4X,3HTOP,4X,3HBOT,4X,3HBOT//22X,
               1 1HI,3X,4HNAME,3X,5HPLWID,2X,5HPLATH,2X,5HPLWID,2X,5HPLATH//)
001026          DO 7101 I = 1,NE
001030          II = NOS(I)
001032     7101 WRITE(6,7102)I,NAME(II),PLWID(I,1),PLATH(I,1),PLWID(I,2),
```

```
                 1(,AIF(I,2)                                                            THDE 219
001054      7102 FORMAT (21X,I2,[illegible],F6.2,X,F6.3,X,F6.2,[illegible])              THDE 220
001054      7200 CONTINUE                                                               THDE 221
001054           RETURN                                                                 THDE 222
001055           END                                                                    THDE 223

SUBPROGRAM LENGTH
001265

UNUSED COMPILER SPACE
037000



                 FUNCTION TOPT (T)                                                      TOPT    1
000003           IF (T .LE. .5) TMIN = .1875                                            TOPT    2
000007           IF (T .GT. .5 .AND. T .LE. .75) TMIN = .25                             TOPT    3
000021           IF (T .GT. .75 .AND. T .LE. 1.5) TMIN = .3125                          TOPT    4
000033           IF (T .GT. 1.5 .AND. T .LE. 2.25) TMIN = .375                          TOPT    5
000045           IF (T .GT. 2.25 .AND. T .LE. 6.0) TMIN = .5                            TOPT    6
000057           TOPT = TMIN                                                            TOPT    7
000061           RETURN                                                                 TOPT    8
000061           END                                                                    TOPT    9

SUBPROGRAM LENGTH
000111

UNUSED COMPILER SPACE
042700



                 SUBROUTINE UNIFST                                                      UNIF    1
            C       DESIGNS THE UNIFORM SECTION GIRDER                                  UNIF    2
000002           COMMON NAME(18),SAREA(18),DEPTH(18),FLWID(18),FLTH(18),WEBTH(18), UNIF    3
                1DAH(18),IX(18),IY(18),NSECT(18),NOS(80),ISTEEL(80),CS(80),             UNIF    4
                2ILCON(80),CL(80),IHCON(80),CH(80),PLATH(80,2),PLWID(80,2),NF,          UNIF    5
                3SLABWO,SLABTH,MR,COOR(81),RTL(5,81,3),CC(5,5),FU,SPAN(4),IOSP(5), UNIF    6
                4TMOE1(80),NA,LENGTH,RO(81),SUBLEN(80),NS,UNLO(9),TRLO(10),PROP(3),UNIF    7
                5FC,STRESS(81,4,2),DESMOM(81,4),DESREA(5,4),DESS(162,4),SFAR(5,3),UNIF    8
                6COST(9),CONSP(16,3,4),COVPL(12,6,2)                                    UNIF    9
000002           COMMON/ONE/JPT3,IPT10                                                  UNIF   10
000002           COMMON/TWO/LPCM                                                        UNIF   11
000002           COMMON/THREE/AS,NCS,IPTA                                               UNIF   12
000002           COMMON/FOUR/NICT,ICOST                                                 UNIF   13
000002           COMMON/SIX/FU                                                          UNIF   14
000002           COMMON/SEVEN/WMIN,CMIN                                                 UNIF   15
000002           COMMON/EIGHT/IST                                                       UNIF   16
000002           COMMON/TEN/HAUNCH                                                      UNIF   17
000002           COMMON/B/SMEFC,ICP                                                     UNIF   18
000002           REAL IX,IY,ISTEEL,ILCON,IHCON,MR,LENGTH                                UNIF   19
000002           DIMENSION TITLE(36), EPS(4), TIL(81,3), STOR(12), LPCM(10)             UNIF   20
```

```
000002   -      DIMENSION TMST(4)                                                UNIF  21
000002          MN = NS - 1                                                      UNIF  22
000004          IF (JPT3 .LT. 9) TSWT01 = SLAMWD / (3. * MN)                     UNIF  23
000012          IF (JPT3 .LT. 9) TSWT02 = SLAMWD / MN                            UNIF  24
000016          IF (MN .EQ. 1) GO TO 4000                                        UNIF  25
        C          DETERMINES THE CRITICAL DESIGN SECTION FOR A CONTINUOUS GIRDER UNIF  26
000020          DSM = 0.0                                                        UNIF  27
000021          ICR = 1                                                          UNIF  28
000022          DO 1500 I = 1, 10, 2                                             UNIF  29
000023          IA = LPCM(I)                                                     UNIF  30
000025          IB = LPCM(I+1)                                                   UNIF  31
000026          IF (IA .EQ. 0 .AND. IB .EQ. 0) GO TO 1500                        UNIF  32
000034          DO 1501 K = IA, IB                                               UNIF  33
000035          DO 1502 J = 1, 4                                                 UNIF  34
000036     1502 TMST(J) = DESMOM(K,J)                                            UNIF  35
000045          IDS = STRMAX(TMST)                                               UNIF  36
000047          IF (ABS(IDS) .LE. DSM) GO TO 1501                                UNIF  37
000052          ICR = K                                                          UNIF  38
000053          DSM = ABS(IDS)                                                   UNIF  39
000054     1501 CONTINUE                                                         UNIF  40
000057     1500 CONTINUE                                                         UNIF  41
000061          SMEFC = DESMOM(ICR,1) + DESMOM(ICR,3) + DESMOM(ICR,4)            UNIF  42
000065          GO TO 1562                                                       UNIF  43
        C          DETERMINES THE CRITICAL DESIGN SECTION FOR A SIMPLE SPAN      UNIF  44
000065     4000 AV = 9999999.                                                    UNIF  45
000067          DT = LENGTH / 2.                                                 UNIF  46
000071          DO 4010 J = 1, NA                                                UNIF  47
000072          IF (ABS(COOR(J) - DT) .GT. AV) GO TO 4010                        UNIF  48
000100          ICR = J                                                          UNIF  49
000100          AV = ABS(COOR(J) - DT)                                           UNIF  50
000102     4010 CONTINUE                                                         UNIF  51
000105          WRITE (6,4030) ICR                                               UNIF  52
000112     4030 FORMAT (1H0,10X,52H0N THE SIMPLE BEAM, THE CRITICAL MOMENT IS AT PUNIF  53
               1OINT ,I2)                                                        UNIF  54
000112          SMEFC = DESMOM(ICR,1)                                            UNIF  55
000114          IF (JPT3 .GT. 9) SMEFC = DESMOM(ICR,1) + DESMOM(ICR,2)           UNIF  56
000121          GO TO 4020                                                       UNIF  57
000122     1562 WRITE(6,1510) ICR                                                UNIF  58
000130     1510 FORMAT (1H0,10X,34HCRITICAL NEGATIVE MOMENT POINT IS ,I2)        UNIF  59
        C          DESIGNS THE CONTINUOUS GIRDER                                 UNIF  60
        C          DETERMINES THE ALLOWABLE STRESSES                             UNIF  61
000130          DO 4200 J = 1, 4                                                 UNIF  62
000132     4200 TMST(J) = DESMOM(ICR, J)                                         UNIF  63
000141          ASFC = BASFAT(TMST,FU,-1000.,PROP(3))                            UNIF  64
000145          ASFT = BASFAT(TMST,FU,1000.,PROP(3))                             UNIF  65
000151          ASF = ASFT                                                       UNIF  66
000152          IF (ABS(ASFC) .LT. ABS(ASFT)) ASF = ABS(ASFC)                    UNIF  67
000157          PLT = 0.0                                                        UNIF  68
000160          PLW = 0.0                                                        UNIF  69
        C          ITERATES THE SECTIONS TO DETERMINE THE MINIMUM WEIGHT SECTION UNIF  70
000161          I = 1                                                            UNIF  71
000162          IST = 0                                                          UNIF  72
000163     1581 IF (I .EQ. 19) GO TO 2603                                        UNIF  73
000165          EFS = DSM * DEPTH(I) / (2. * IX(I))                              UNIF  74
000171          ASTR = ALLOW(DESMOM(ICR,1),FLWID(I),LPCM,IOSP,ICR,NS,PROP,COOR,  UNIF  75
               1DESMOM,SUBLEN,FU,1)                                              UNIF  76
000206          IF (ABS(ASF) .LT. ABS(ASTR)) ASTR = ABS(ASF)                     UNIF  77
000213          IF (ABS(EFS) .GT. ASTR) GO TO 1580                               UNIF  78
```

```
000220             IST = 1                                                      UNIF  79
000221             I = I + 1                                                    UNIF  80
000222             GO TO 1081                                                   UNIF  81
000223        4020 CONTINUE                                                     UNIF  82
          C              DESIGNS THE SIMPLE SPAN PROBLEM                        UNIF  83
          C              ITERATES THE SECTIONS TO DETERMINE THE MINIMUM WEIGHT DESIGN UNIF  84
000223             I = 1                                                        UNIF  85
000224             IST = 0                                                      UNIF  86
000225        4060 IF (I .EQ. 19) GO TO 2603                                    UNIF  87
000227             SIS = IS(I)                                                  UNIF  88
000231             SC = DEPTH(I) / 2.                                           UNIF  89
000233             IF (JPTS .GT. 4) GO TO 700                                   UNIF  90
000236             CALL ICALC(ISWID1,SLABTH,0.,0.,DEPTH(I),SAREA(I),SIS,SC,HAUNCH, UNIF  91
                  1CLC,SLC)                                                     UNIF  92
000250             CALL ICALC(ISWID2,SLABTH,0.,0.,DEPTH(I),SAREA(I),SIS,SC,HAUNCH, UNIF  93
                  1CHC,SHC)                                                     UNIF  94
000263             GO TO 701                                                    UNIF  95
000264         700 CHC = SC                                                     UNIF  96
000266             SHC = SIS                                                    UNIF  97
000267             CLC = SC                                                     UNIF  98
000270             SLC = SIS                                                    UNIF  99
          C              DETERMINES THE ALLOWABLE STRESSES                      UNIF 100
000271         701 TMST(1) = DESMOM(ICR,1) * SC / SIS                           UNIF 101
000274             TMST(2) = DESMOM(ICR,2) * CHC / SHC                          UNIF 102
000277             TMST(3) = DESMOM(ICR,3) * CLC / SLC                          UNIF 103
000301             TMST(4) = DESMOM(ICR,4) * CHC / SHC                          UNIF 104
000304             TDS = STRMAX(TMST)                                           UNIF 105
000306             ASF = HASFAT(TMST,FJ,1000.,PROP(3))                          UNIF 106
000312             ASTR = PROP(1)                                               UNIF 107
000313             IF (ASF .LT. PROP(1)) ASTR = ASF                             UNIF 108
000317             IF (ABS(TDS) .GT. ASTR) GO TO 1580                           UNIF 109
000324             IST = I                                                      UNIF 110
000325             I = I + 1                                                    UNIF 111
000326             GO TO 4060                                                   UNIF 112
000327        1580 IF (IST .EQ. 0) GO TO 2605                                   UNIF 113
000330        2603 IF (IPTL .GE. 1) WRITE (6,1588) NAME(IST)                    UNIF 114
000341        1588 FORMAT (1H0,20X,8HUSE THE ,A7,23H FOR A UNIFORM SECTION.////) UNIF 115
          C              SETS UP THE DESIGN ARRAYS                              UNIF 116
000341             DO 1530 J = 1, NF                                            UNIF 117
000343             PLATH(J,1) = PLT                                             UNIF 118
000345             PLATH(J,2) = PLT                                             UNIF 119
000346             PLWID(J,1) = PLW                                             UNIF 120
000347             PLWID(J,2) = PLW                                             UNIF 121
000350        1530 NOS(J) = IST                                                 UNIF 122
000354             CALL SETIC                                                   UNIF 123
000355             GO TO 2602                                                   UNIF 124
000356        2605 WRITE (6,2606)                                               UNIF 125
000362        2606 FORMAT (1H0,42HNO SECTION WITHOUT PLATES IS SATISFACTORY.)   UNIF 126
000362             RETURN                                                       UNIF 127
000363        2602 DO 3000 I = 1, 12                                            UNIF 128
000365             DO 3000 J = 1, 6                                             UNIF 129
000366             DO 3000 K = 1, 2                                             UNIF 130
000367        3000 COVPL(I,J,K) = 0.0                                           UNIF 131
000403             CALL CRSHRS                                                  UNIF 132
          C              DESIGNS THE BEARING STIFFENERS                         UNIF 133
000404             CALL STIFF(IOSP,IS,DESREA,NOS,FLTH,FLWID,WEBTH, EPTH,PLCO,FM, UNIF 134
                  1FJ)                                                          UNIF 135
                         DESIGNS THE SHEAR CONNECTORS                           UNIF 136
```

```
000417          CALL COSTS                                                      UNIF 137
          C         DETERMINE THE VALUES OF THE OBJECTIVE FUNCTION              UNIF 138
000420          CALL OBJECT                                                     UNIF 139
          C         KEEP THE DESIGN AS A STARTING POINT                         UNIF 140
000421          CALL KEEP(NF,CUS,PLATH,PLWID,COVPL,CMIN,WT ,IPT6)               UNIF 141
          C         PRINTOUT THE DESIGN INFORMATION                             UNIF 142
000431          IF (IFTIC .LE. 1) GO TO 2611                                    UNIF 143
000434          CALL PRSEC(NF,NOS,NAME,PLWID,PLATH,ISTEEL,CS,ILCON,CL,IHCON,CH, UNIF 144
               1COMX,SUBLEN,C,NCY)                                              UNIF 145
000452          CALL PRCOV (COVPL)                                              UNIF 146
000454          WRITE (6,100)                                                   UNIF 147
000460      100 FORMAT (1H1)                                                    UNIF 148
000460          CALL PRSTIF(NS,HEAD)                                            UNIF 149
000462          CALL PRCON(NS,CONSP,NCS)                                        UNIF 150
000465          WRITE (6,1518) WTOT, TCOST                                      UNIF 151
000475     1518 FORMAT (1H0,10X,15HTOTAL WEIGHT = ,F15.5,5X,13HTOTAL COST = ,   UNIF 152
               1F15.2)                                                          UNIF 153
000475          IF (IPT6 .GT. .5) GO TO 2610                                    UNIF 154
000501          WRITE (6,1582) CMIN                                             UNIF 155
000506     1582 FORMAT (1H0,30X,7HCMIN = ,F12.2)                                UNIF 156
000506          GO TO 2611                                                      UNIF 157
000507     2610 WRITE (6,3582) WMIN                                             UNIF 158
000515     3582 FORMAT (1H0,30X,7HWMIN = ,F12.2)                                UNIF 159
000515     2611 CONTINUE                                                        UNIF 160
000515          RETURN                                                          UNIF 161
000516          END                                                             UNIF 162

SUBPROGRAM LENGTH
001343


UNUSED COMPILER SPACE
040100
```